



VNiVERSiDAD  
D SALAMANCA

UNIVERSIDAD DE SALAMANCA

Departamento de Informática y Automática

UN SISTEMA DE  
ODOMETRÍA VISUAL MONOCULAR  
BAJO RESTRICCIONES  
DE TIEMPO REAL

TESIS DOCTORAL

TESIS DOCTORAL PRESENTADA POR:

D. JESÚS FERNANDO RODRÍGUEZ-ARAGÓN

**Dirigida por:**

DR. D. VIDAL MORENO RODILLA  
DRA. DÑA. BELÉN CURTO DIEGO

Julio 2015



## **Agradecimiento Oficial**

Esta Tesis Doctoral se ha realizado bajo el amparo y la financiación del Programa de Ayuda a la Contratación de Personal Investigador de Reciente Titulación (Formación de Personal Investigador). Orden EDU/1933/2008, de 11 de noviembre.



**VNiVERSiDAD  
D SALAMANCA**

**UNIVERSIDAD DE SALAMANCA**  
**Departamento de Informática y Automática**

**UN SISTEMA DE  
ODOMETRÍA VISUAL MONOCULAR  
BAJO RESTRICCIONES  
DE TIEMPO REAL**

TESIS DOCTORAL PRESENTADA POR:  
D. JESÚS FERNANDO RODRÍGUEZ-ARAGÓN

**Dirigida por:**  
DR. D. VIDAL MORENO RODILLA  
DRA. DÑA. BELÉN CURTO DIEGO

El doctorando

Salamanca, Julio de 2015





Vidal Moreno Rodilla, *Profesor Titular de Universidad del Departamento de Informática y Automática de la Universidad de Salamanca*

**HACE CONSTAR:** *Que D. Jesús Fernando Rodríguez-Aragón, Ingeniero Informático por la Universidad de Salamanca ha realizado bajo mi dirección la Memoria que lleva por título UN SISTEMA DE ODOMETRÍA VISUAL MONOCULAR BAJO RESTRICCIONES DE TIEMPO REAL, con el fin de obtener el grado de Doctor por la Universidad de Salamanca.*

Y para que surta los efectos oportunos firmo en Salamanca, a doce de Mayo de dos mil quince.



*A mi familia, porque se lo debo todo.*



# Agradecimientos

Resumir en una hoja las personas que han ido pasando por tu vida durante los años de realización de la Tesis genera, cuando menos, cierta nostalgia ya no sólo por el trabajo realizado, sino también por la vida que se ha vivido alrededor de esta Tesis.

Por supuesto, lo primero que me viene a la mente es mi familia. Sin su apoyo incondicional literalmente no habría terminado esta tesis. Estar rodeado de tantos doctores al final hace sentirse a uno un poco como Howard Wolowitz y la fuerza de voluntad -que bien saben ellos que en ocasiones escasea en mi carácter- que me inculcaron en cada momento, ha hecho que al final este día haya llegado. Mi padre y mi madre han logrado que hoy esté escribiendo estas líneas, orgulloso del trabajo hecho y orgulloso y agradecido por el cariño y también, por qué no decirlo, la exigencia que siempre me han mostrado, que bien compaginada, suele resultar la mejor lección y el mejor apoyo que unos padres pueden dar. Porque estoy donde estoy gracias a ellos, gracias a que me han enseñado lo que vale cada esfuerzo y cómo se disfruta después el trabajo bien hecho.

Mi hermano, que ha estado ahí mostrando una ayuda vital tanto en los momentos en los que parecía que “todo salía”, como en los momentos en los que los resultados no eran del todo bonitos; porque me ha aportado su experiencia, porque me ha ido encauzando cuando mi camino se iba torciendo, porque ha sido capaz de motivar en los momentos de dudas y capaz de alegrarse en los momentos más optimistas. Mi cuñada, con quien he compartido, en tiempo y forma, época de investigación y escritura, mirándonos mutuamente con cara de sufrimiento e ilusión común por supuesto. Y cómo no, también mi sobrina y ahijada, que me ha acompañado durante el último año sacándome siempre una sonrisa.

Y dentro de mi familia, qué decir de aquella chica con quien -cuando empecé esta Tesis- simplemente compartía pasillo de departamento y leves saludos cuando nos cruzábamos. Esa chica que durante el ecuador de esta Tesis pasó a ser mi novia; y a día de hoy, escribiendo estas líneas, es mi mujer. Sin ella, sin su amor, sin su alegría, sin su sutil tenacidad, sin su apoyo y sin su profesionalidad, no sería lo que soy. Porque veo que la vida me ha dado el mejor de los regalos, permitirme estar a su lado, aprendiendo y disfrutando cada día de su forma de ser, su vitalidad y su amor.

Sin olvidarme de todos y cada uno de los miembros de mi familia, tíos y primos, que siempre han estado ahí y de los que siempre he sentido su apoyo y su cariño.

Tamén miña nova familia, sempre chea de vitalidade e ledicia, que me acolleu dende o primeiro momento coma un máis, sentíndome xa un galego máis e que fan que o meu día sexa aínda máis feliz se cabe; que engaden un sorriso a cada minuto e que fixeron que a motivación medrara mostrándome o seu lado máis amable. Porque o feito de espertarse cada día un pouco máis feliz fai que os días teñan máis de 24 horas.

Mis tutores de Tesis, Belén y Vidal, a quienes a día de hoy considero mis amigos, porque logran un ambiente de trabajo excepcional que favorece la motivación y a creer en lo que estás haciendo, porque desde el principio alimentaron mi amor por la investigación y la ciencia. Mis compañeros de facultad, especialmente Raúl y Vicen, porque han aguantado muchos cafés con días peores y días mejores, pero que han servido de base y sostén para que siguiera siempre hacia adelante.

Y qué sería del quehacer diario sin un buen grupo de amigos con quien compartir alegrías y penas. Mis amigos de la infancia, que mantenemos una amistad forjada con el paso de los años y con quienes sé que puedo contar en cualquier momento. Forman esa red de seguridad que proporciona la tranquilidad de saberte siempre bien acompañado en esta vida. Mis amigos de la facultad, con quienes siempre ha sido fácil disfrutar de buenos momentos, que logran fácilmente una carga de energía nueva para afrontar nuevos retos.

Los nuevos amigos que he hecho durante este tiempo, personas que hacen que sea fácil dedicarse en cuerpo y alma al trabajo; porque uno sabe que tras un día productivo, siempre habrá tiempo para pasar un buen rato con ellos, ya sea junto a una caña en el “amarre” en una casa

rural -donde sea-, en un concierto, un festival quizá, alrededor de un balón de fútbol -juguemos nosotros o jueguen otros-, etc. Porque sin ellos, mi vida sería menos vida, porque esta Tesis va ligada a ellos. A veces pienso que en vez de en los Agradecimientos, deberían aparecer como un apartado más dentro de los Resultados y Conclusiones de esta Tesis. Porque para el resto de mis días, yo seguiré tomando el “pincho a las once”...

Un gracias enorme que quiero que viaje a un pueblecito zamorano cerca de la frontera, porque dos de sus habitantes me han reconciliado muchas veces con el mundo. Porque sería otra persona de no haberlas conocido; porque sus cafés, sus cañas, sus pinchos, sus cenas han sido el catalizador necesario durante estos años. Porque nunca un “¿te hace una caña?” fue tan sinceramente ofrecido y tan felizmente aceptado.

Y un gracias especial a Jason. Porque tras estos años siendo uña y carne, pasando más horas juntos que con nuestras propias familias, uno descubre que con quien mejor se trabaja es con un verdadero amigo. Porque desde estas líneas quiero mostrar mi aprecio y mi respeto hacia todo lo que hace, mi aprecio y mi respeto hacia todo lo que ayuda; pero también, y más importante, quiero mostrarle mi amistad eterna, mi agradecimiento infinito y mi cariño.

Por último, y no menos importante, agradecer a todas y cada una de las personas que forman parte de mi vida, porque, como empezaba estos agradecimientos, la realización de una Tesis acaba diluyéndose en el día a día hasta el punto de no existir ninguna línea que separe el trabajo del ocio, sino que se acaba fundiendo todo en uno; y cada persona cumple una función vital dentro de este tiempo.

Gracias.





# Resumen

La localización en cada instante de tiempo es esencial para un robot móvil autónomo. Es un problema que ha sido tratado desde múltiples perspectivas. A la hora de afrontar los problemas de posicionamiento, es común la utilización de diferentes sensores que proporcionan información directa sobre la localización actual del robot o bien sobre los cambios que se han producido en el entorno del mismo.

La odometría visual es el proceso mediante el que se calcula una estimación del movimiento realizado por un robot móvil o por un vehículo a partir de los datos capturados por el sistema de visión del mismo.

Actualmente, la mayor parte de los algoritmos de odometría visual propuestos no trabajan en tiempo de ejecución, sino que los resultados se obtienen mayoritariamente *offline*, esto es, se toman las imágenes y posteriormente se realiza el procesamiento de las mismas. Otros trabajan en una frecuencia de toma de imágenes muy baja, lo que implica que se precisa un movimiento del robot muy lento. Esto es debido a que los algoritmos actuales de odometría visual conllevan cálculos muy pesados y de alta carga computacional. Esto implica que para su correcta ejecución, muchas veces es necesario disponer de equipos informáticos de alto nivel, con el coste que ello conlleva. Los sistemas de visión que se utilizan suelen ser sistemas de visión avanzados de alto coste.

Muchos de los trabajos propuestos con anterioridad se basan en el cálculo del movimiento a partir de la estructura, esto es, el cálculo de la estructura en 3-D a partir de una secuencia de imágenes capturadas en diferentes instantes de tiempo y, con ello, el cálculo del movimiento realizado. En cambio, el trabajo que se presenta en esta Tesis basa la estimación del movimiento que ha tenido lugar calculándolo como un giro sobre un punto denominado Centro Instantáneo de Rotación. Los

trabajos previos necesitan en muchas ocasiones de sensores internos para calcular el movimiento realizado por el robot, ya sea para calcular la velocidad del robot, o para calcular los giros que han tenido lugar.

Se presenta en esta Tesis un procedimiento de odometría visual puramente exteroceptivo fiable y robusto capaz de realizar el cálculo del movimiento de un robot a partir de las imágenes capturadas por una única cámara que se encuentra sobre el robot. Este procedimiento resulta ser viable para ejecutarse en tiempo real durante la navegación del robot haciendo uso de elementos computacionales domésticos.

A modo de caso de uso, se presenta también el diseño y el desarrollo de un sistema de localización basado en el filtro extendido de Kalman (EKF), que integra conjuntamente el sistema de odometría visual con el sistema de odometría de las ruedas, logrando de esta manera un sistema odométrico más robusto y fiable en diferentes entornos.

# Publicaciones

## Artículos en revistas

SERRANO, F.J.; RODRÍGUEZ-ARAGÓN, J.F.; CURTO, B. Y MORENO, V.

“Multi-Agent Cooperation for Advanced Teleoperation of an Industrial Forklift in Real-Time Environment”

*Advances on Practical Applications of Agents and Multi-agent Systems* 88, 57-62, (2011).

FERNÁNDEZ-CARAMÉS, C.; MORENO, V.; CURTO, B.; RODRÍGUEZ-ARAGÓN, J.F. Y SERRANO, F.J.

“A Real-time Door Detection System for Domestic Robotic Navigation”

*Journal of Intelligent & Robotic Systems* 76, 1, 119-136, (2014).

PITTI PATINO, K.; CURTO, B.; MORENO, V.; RODRÍGUEZ CONDE, M.J. Y RODRÍGUEZ-ARAGÓN, J.F.

“Using Robotics as a Learning Tool in Latin America and Spain”

*IEEE Revista Iberoamericana de Tecnologías del Aprendizaje* 9, 4, 144-150, (2014).

BLANCO, F.J.; CURTO, B.; MORENO, V.; PITTI PATINO, K. Y RODRÍGUEZ-ARAGÓN, J.F.

“Improving Robotics Teaching in the Computer Engineering Degree through the Action-Research Approach”

*Innovative Methods of Teaching Engineering Special Issue, International Journal of Engineering Education* 31, 3, 798-804, (2015).

## Congresos Internacionales

RODRÍGUEZ-ARAGÓN, J.F.; MORENO, V.; CURTO, B.; SERRANO, F.J.; ALVES SANTOS, R. Y POLO, M.J. "Indoor Location Using Wireless Networks Based on Bayesian Reasoning" *6th International Conference on Web Information Systems and Technologies*, Valencia, España, Abril 2010.

SERRANO, F.J.; GONZÁLEZ TALAVÁN, G.; MORENO, V.; CURTO, B.; RODRÍGUEZ-ARAGÓN, J.F. Y MORENO, A.M.  
"VoIP Integration: VoIP Control and Processing System"  
*6th International Conference on Web Information Systems and Technologies*, Valencia, España, Abril 2010.

SERRANO, F.J.; RODRÍGUEZ-ARAGÓN, J.F.; CURTO, B. Y MORENO, V.  
"Multi-Agent Cooperation for Advanced Teleoperation of an Industrial Forklift in Real-Time Environment"  
*9th International Conference on Practical Applications of Agents and Multi-Agent Systems*, Salamanca, España, Abril 2011.

## Congresos Nacionales

RODRÍGUEZ-ARAGÓN, J.F.; MORENO, V.; CURTO, B.; SERRANO, F.J. Y ALVES SANTOS, R.  
"Localización en interiores con redes wireless"  
*XXX Jornadas de Automática*, Valladolid, España, Septiembre 2009.

SERRANO, F.J.; CURTO, B.; MORENO, V.; RODRÍGUEZ-ARAGÓN, J.F. Y ALVES SANTOS, R.  
"Sistema en tiempo real para el procesamiento y control de VoIP"  
*XXX Jornadas de Automática*, Valladolid, España, Septiembre 2009.

ALVES SANTOS, R.; CURTO, B.; MORENO, V.; RODRÍGUEZ-ARAGÓN, J.F. Y BLANCO, F.J.  
"Control y Supervisión de Plantas Didácticas vÃa OPC con EDUSCA"  
*XXX Jornadas de Automática*, Valladolid, España, Septiembre 2009.

ALVES SANTOS, R.; CURTO, B.; MORENO, V.; VICENTE, J.A. Y RODRÍGUEZ-ARAGÓN, J.F.

“Sistema distribuido heterogéneo para la automatización de una carretilla”

*XXXI Jornadas de Automática*, Jaén, España, Septiembre 2010.

FERNÁNDEZ-CARAMÉS, C.; MORENO, V.; CURTO, B.; ALVES SANTOS, R. Y RODRÍGUEZ-ARAGÓN, J.F.

“Controlador Borroso de Seguimiento de Segmentos para Diversas Configuraciones Cinemáticas”

*XXXI Jornadas de Automática*, Jaén, España, Septiembre 2010.

RODRÍGUEZ-ARAGÓN, J.F.; CURTO, B.; MORENO, V.; FERNÁNDEZ-CARAMÉS, C. Y SERRANO, F.J.

“Técnicas de Visión Artificial para Estimar el Estado de un Robot”

*XXXI Jornadas de Automática*, Jaén, España, Septiembre 2010.

## Artículos en revistas pendientes de aceptación

RODRÍGUEZ-ARAGÓN, J.F.; MORENO, V.; CURTO, B.; SERRANO, F.J.; ALVES SANTOS, R. Y FERNÁNDEZ-CARAMÉS, C.

“Real-time Mobile Robot Localization Using Monocular Visual Odometry”

Enviado a: *Robotica (Cambridge University Press)* (2015).



# Índice general

<b>1. Introducción</b>	<b>31</b>
1.1. Orígenes de la odometría visual . . . . .	34
1.2. Motivación . . . . .	40
1.3. Objetivos . . . . .	41
1.4. Metodología . . . . .	43
1.5. Estructura del documento . . . . .	45
<b>2. Estado del arte</b>	<b>49</b>
2.1. Odometría visual estéreo . . . . .	50
2.2. Odometría visual monocular . . . . .	57
2.2.1. Métodos basados en características . . . . .	59
2.2.2. Métodos globales . . . . .	60
2.2.3. Métodos híbridos . . . . .	62
2.3. Optimización en el cálculo de la odometría visual . . . . .	63
2.4. Cálculo del movimiento . . . . .	65
2.4.1. Detección y seguimiento de características . . . . .	66
2.4.2. Eliminar valores anómalos . . . . .	75
2.5. Consideraciones finales . . . . .	83



<b>3. Procedimiento de Odometría Visual propuesto</b>	<b>87</b>
3.1. Introducción . . . . .	87
3.2. Modelo teórico para el desarrollo . . . . .	89
3.3. Detección y seguimiento de características . . . . .	95
3.3.1. Detección de puntos característicos en una imagen	95
3.3.2. Seguimiento de las características . . . . .	98
3.4. Cálculo del estado . . . . .	100
3.4.1. Proyección de un punto de la imagen . . . . .	101
3.4.2. Eliminar valores anómalos . . . . .	103
3.4.3. Cálculo del Centro Instantáneo de Rotación . . . .	104
3.4.4. Filtrado de resultados . . . . .	109
3.4.5. Clasificador borroso . . . . .	111
3.4.6. Cálculo de la nueva posición . . . . .	116
3.5. Consideraciones finales . . . . .	118
<b>4. Resultados</b>	<b>121</b>
4.1. Introducción . . . . .	121
4.2. Resultados del procedimiento de Odometría Visual . . . .	124
4.2.1. Resultados en interiores . . . . .	124
4.2.2. Resultados en exteriores . . . . .	135
4.3. Adecuación a la ejecución en tiempo real . . . . .	142
4.4. Caso práctico: Experiencia de fusión sensorial . . . . .	144
4.4.1. Implementación del EKF . . . . .	147
4.4.2. Resultados de la Fusión Sensorial . . . . .	152
<b>5. Conclusiones</b>	<b>161</b>
5.1. Conclusiones y principales contribuciones . . . . .	161

5.2. Trabajo futuro . . . . .	167
<b>A. Base teórica de la fusión sensorial</b>	<b>169</b>
A.1. El Filtro de Kalman . . . . .	169
A.1.1. Suposiciones . . . . .	171
A.1.2. Formulación del problema de estimación . . . . .	176
A.1.3. Algoritmo . . . . .	177
A.2. El Filtro de Kalman Extendido (EKF) . . . . .	180
A.2.1. Modelo del Sistema . . . . .	181
A.2.2. Modelo de Observación . . . . .	187



# Índice de tablas

4.1. Tiempo de ejecución del procedimiento VO (en milisegundos) . . . . .	143
---	-----



# Índice de figuras

1.1. Robot utilizado en el trabajo de Moravec y su sistema de visión móvil . . . . .	37
1.2. Robot utilizado en el trabajo de Lacroix et al. . . . .	39
2.1. Resultados del trabajo de Olson et al. . . . .	52
2.2. Resultados del trabajo de Cheng et al. . . . .	54
2.3. Imágenes relativas al trabajo de Milella y Siegwart. a)Puntos característicos detectados en dos imágenes consecutivas. b)Seguimiento de las características y filtrado de las mismas. c) Robot utilizado . . . . .	56
2.4. Comport: tensor cuadrifocal . . . . .	58
2.5. Creación del mapa en 3-D por Tardif . . . . .	61
2.6. Comparación de cálculo de odometría visual utilizando un método híbrido (global y basado en características) y un método basado únicamente en características, por Scaramuzza . . . . .	63
2.7. Comparativa de los detectores de características: propiedades y rendimiento, por Scaramuzza . . . . .	69
2.8. Ejemplo de una característica y el descriptor SIFT . . . . .	70
2.9. Scaramuzza: cálculo del Centro Instantáneo de Rotación en un giro . . . . .	81
2.10. Scaramuzza: votación en el histograma de ángulos . . . . .	82

3.1.	Situación de la cámara . . . . .	90
3.2.	Orientación robocéntrica estática del suelo . . . . .	91
3.3.	Cálculo del Centro Instantáneo de Rotación . . . . .	94
3.4.	a)Vectores de desplazamiento para un par de imágenes,b)Puntos CIR candidatos . . . . .	106
3.5.	Histograma de CIR candidatos . . . . .	108
3.6.	CIR filtrado (rojo) y CIR sin filtrar (azul) . . . . .	111
3.7.	Avance en recta . . . . .	112
3.8.	Entradas del clasificador borroso: a)Variabilidad del centro de rotación en los instantes anteriores, b)Tipo de movimiento en los instantes anteriores . . . . .	114
3.9.	Salida del clasificador borroso . . . . .	115
3.10.	Reglas del clasificador borroso . . . . .	116
4.1.	Plataforma ArViBOT . . . . .	123
4.2.	a) Baldosas del Hall de la Facultad con brillos. b) Baldosas de los pasillos. c) Baldosas de los pasillos con sombras y brillos. d) Aparición de personas caminando en la captura de la imagen. . . . .	126
4.3.	Resultados del procedimiento de odometría visual en un movimiento cíclico en condiciones idóneas. Verde: movimiento real. Azul: odometría de las ruedas (WO). Rojo: Odometría visual (VO). . . . .	128
4.4.	Distancia (en milímetros) entre el punto real y el punto calculado por el procedimiento de odometría visual (rojo) y la odometría de las ruedas (azul). . . . .	129
4.5.	a) Imagen con mucha iluminación. b) Imagen con poca iluminación. c) Imagen desenfocada. d) Imagen desenfocada.131	
4.6.	Resultados del procedimiento de cálculo de odometría visual. Verde: movimiento real. Azul: Odometría de las ruedas. Rojo: Odometría visual . . . . .	132

4.7. Distancia (en mm) entre el punto real y los puntos calculados por la odometría visual (rojo) y por la odometría de las ruedas (azul) . . . . .	134
4.8. a) El robot durante la navegación. b) Suelo abrupto en exteriores. c) Un salto en el suelo. d) Suelo en exteriores menos texturizado . . . . .	136
4.9. Resultados del movimiento cíclico en exteriores. Verde: movimiento real. Azul: Odometría de las ruedas. Rojo: Odometría visual . . . . .	137
4.10. Distancia (en mm) entre el punto real y los puntos calculados por la odometría visual (rojo) y por la odometría de las ruedas (azul) . . . . .	139
4.11. Recorrido realizado en exteriores. Verde: movimiento real. Azul: Odometría de las ruedas. Rojo: Odometría visual . . . . .	140
4.12. Distancia (en mm) entre el punto real y los puntos calculados por la odometría visual (rojo) y por la odometría de las ruedas (azul) . . . . .	141
4.13. Resultados de la fusión sensorial. Azul: Odometría de las ruedas. Rojo: Odometría visual. Negro: Fusión sensorial. Verde: Trayectoria real . . . . .	156
4.14. Resultados de la fusión sensorial. Azul: Odometría de las ruedas. Rojo: Odometría visual. Negro: Fusión sensorial. Verde: Trayectoria real . . . . .	158
A.1. Relación entre el sistema de referencia local del robot y el sistema de referencia global . . . . .	184





# Capítulo 1

## Introducción

La localización en cada instante de tiempo es esencial para un robot móvil autónomo. Es un problema que ha sido tratado desde múltiples perspectivas. A la hora de afrontar los problemas de posicionamiento, es común la utilización de diferentes sensores que proporcionan información directa sobre la localización actual del robot o bien sobre los cambios que se han producido en el entorno del mismo.

En entornos donde se encuentra disponible una señal GPS, los sensores GPS proporcionan información muy detallada con un margen de error reducido [38]. Sin embargo, incluso en ese tipo de entornos, se utilizan otros sensores para ajustar y mejorar el cálculo de la posición del robot puesto que en situaciones no ideales, los sensores GPS proporcionan medidas con error. Ejemplos de este tipo de situaciones son entornos urbanos [79] con altos edificios, bosques frondosos, etc. Para solucionar este tipo de problemas, se plantean sistemas que utilizan balizas, que permiten una localización global y que proporcionan nueva información para un mejor cálculo de la posición actual del robot [4, 10]. Por otra parte, en la literatura, se pueden encontrar múltiples trabajos de fusión sensorial cuyo objetivo es un mejor cálculo de la posición del robot en cada instante de tiempo utilizando varias fuentes de información [112, 22, 102, 40].

En entornos en los que no se dispone de señal GPS, el problema del cálculo de la posición de un robot se torna más complejo al no existir un sensor preciso. Dentro de este tipo de entornos, podemos hablar de navegación en interiores, vehículos aéreos o bajo el agua. La localización en interiores es un problema aún en estudio. Existen diferentes métodos

de localización que tienen un funcionamiento aceptable en determinadas condiciones, pero que introducen mucho error en situaciones no ideales. En este sentido, se han propuesto técnicas para la localización en un edificio público basadas en balizas, como puede ser utilizando un mapa de potencias de señales WiFi [36, 54, 84, 37], sensores geomagnéticos [12] o sensores láser junto con técnicas de visión artificial [18]. Sin embargo, este tipo de técnicas, que funcionan correctamente en determinadas situaciones, se encuentran con problemas cuando cambian las condiciones de trabajo. Por ejemplo, en una localización dentro de un edificio público, los resultados dependerán de la hora del día, es decir, si el edificio está vacío o si en el edificio hay muchas personas caminando por sus pasillos.

De la misma manera, el proceso de localización tampoco es trivial para robots que realizan movimientos en entornos heterogéneos, esto es, que su entorno es una mezcla de movimientos en interiores y movimientos en exteriores. Por tanto, la idea de disponer de un método capaz de operar en ambos entornos resulta ser de gran utilidad y necesidad.

La odometría visual es el proceso mediante el que se calcula una estimación del movimiento realizado por un robot móvil o por un vehículo a partir de los datos capturados por el sistema de visión del mismo. El término de odometría visual (OV) fue utilizado por primera vez en el año 2004 por Nister [74]. El concepto de estimar el movimiento a partir de los datos adquiridos con un sistema de visión es similar al concepto de odometría de las ruedas. Para calcular la nueva posición de un robot móvil utilizando los datos del movimiento de las ruedas se necesita conocer la posición del robot en el instante de tiempo anterior y el incremento de los *encoders* de ambas ruedas que se ha producido entre el instante de tiempo anterior y el actual. En el caso de OV, para calcular la nueva posición utilizando los datos del sistema de visión de un robot, se necesita conocer la posición del robot en el instante de tiempo anterior y una estimación del movimiento realizado, a partir de las diferencias entre las imágenes capturadas en el instante de tiempo anterior y el actual.

Al ser ambos, sistemas de localización incrementales, el error producido durante una estimación no se corrige en ningún momento sino que es acumulado. De esta manera, utilizados de forma aislada, y sin corrección posible, estos sistemas de localización propagan el error aumentando éste con el paso del tiempo [6]. Los sistemas de localización basados en cálculos odométricos se pueden utilizar junto con otros sistemas de localización que corrijan, o al menos minimicen los posibles errores que

se producen en el cálculo del movimiento a partir de los datos de la odometría.

La OV examina los cambios producidos entre dos imágenes capturadas en dos instantes de tiempo para así estimar el movimiento que ha tenido lugar entre ambos instantes de tiempo. El problema de la estimación del movimiento utilizando únicamente los sistemas de visión data de los últimos treinta años. Es durante los últimos diez años cuando se aportan soluciones con posibilidad de ejecución en problemas de tiempo real.

## 1.1. Orígenes de la odometría visual

El problema del cálculo de la posición relativa de la cámara respecto al entorno y la estructura en 3-D del mismo a partir de un conjunto de imágenes se conoce como estructura a partir del movimiento, *Structure From Motion* (SFM) [55, 34]. SFM trata el problema de obtener información en 3-D a partir de una secuencia de imágenes en 2-D. La odometría visual es, en definitiva, un caso particular de SFM, ya que calcula el movimiento propio de una o más cámaras a partir de las imágenes capturadas en diferentes instantes de tiempo.

De alguna manera, el pionero en la detección del movimiento propio, es decir, el movimiento de un sistema de visión que se encuentra sobre un robot móvil, fue Moravec en su tesis en el año 1980 [67]. Su trabajo se centra en la detección de obstáculos durante la navegación, a partir de los datos obtenidos por un sistema de captura de imágenes. Se calcula la posición en 3-D de cada objeto encontrado en el campo de visión a partir de la secuencia de imágenes en 2-D. Incorpora uno de los primeros detectores de características, que denomina *Operador de Interés*, y que a día de hoy se conoce como "Detector de Esquinas de Moravec"[68]. Posteriormente, se han desarrollado nuevos detectores de esquinas, Forstner [21] y Harris y Stephens [34, 33].

En el trabajo de Moravec se utiliza un robot móvil que transporta un sistema de visión que se puede mover horizontalmente sobre un rail. Cada cierto tiempo (avanzaba 1 metro cada 5 minutos) el robot se detenía, tomaba una serie de nueve imágenes mientras la cámara se movía sobre el rail en intervalos equidistantes. El procedimiento propuesto en el trabajo de Moravec consistía en:

- calcular las esquinas en cada una de las imágenes y encontrar los puntos coincidentes respecto al resto de las ocho imágenes
- buscar también las posibles coincidencias con las posibles posiciones del robot en el estado anterior
- eliminar los valores anómalos debido a inconsistencias por la profundidad de los puntos, utilizando las ocho posibles parejas de imágenes estéreo
- calcular el movimiento realizado y las posiciones relativas de los obstáculos mediante la transformación rígida o isometría para alinear los puntos 3-D vistos en dos posiciones consecutivas del robot

Esto generaba un sistema de ecuaciones que se resolvía mediante mínimos cuadrados ponderados, donde los pesos eran inversamente proporcionales a la distancia al punto 3-D. La figura 1.1, obtenida directamente de la tesis de Moravec [67], muestra el robot utilizado durante el trabajo y el sistema de visión sobre el rail utilizado.

El procedimiento propuesto por Moravec se sigue utilizando a día de hoy en los trabajos de odometría visual: captura de imágenes, detección de características, seguimiento o búsqueda de coincidencias de las características en la secuencia temporal de imágenes, eliminación de valores anómalos y cálculo del movimiento realizado o, lo que es lo mismo, de la estructura en 3-D del entorno.

Aunque Moravec utiliza una única cámara, al ser ésta móvil y tomar nueve imágenes en cada localización del robot, se puede considerar que el trabajo de Moravec utiliza un sistema de visión estéreo, ya que se conoce la distancia que se mueve la cámara entre dos imágenes distintas en la misma localización del robot. Por tanto, se puede hablar de 8 parejas posibles de imágenes estéreo por cada localización del robot, y así se utilizan durante el desarrollo del trabajo.

Durante los últimos 20 años del siglo XX, los trabajos de odometría visual tenían como objetivo la localización de robots móviles en entornos espaciales. Es decir, muchos de los trabajos tienen relación con los vehículos no tripulados enviados por la NASA a otros planetas, *rovers*, básicamente aquéllos encuadrados en el programa de exploración de Marte [67, 61, 62, 50, 76]. El objetivo era calcular el movimiento realizado por un robot móvil en un entorno accidentado, sin balizamiento, con

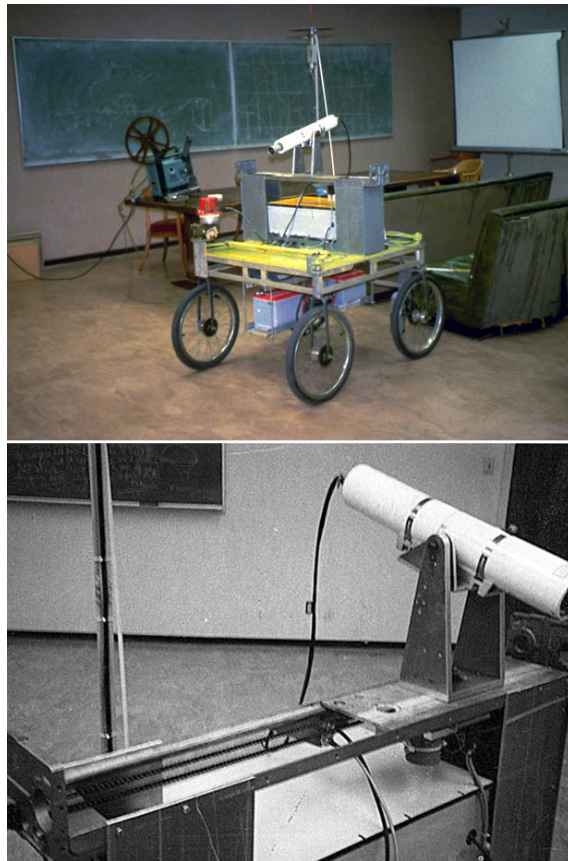


Figura 1.1: Robot utilizado en el trabajo de Moravec y su sistema de visión móvil

posibles derrapes, etc. En definitiva, se trataba de abordar todos los problemas que podrían encontrarse en la localización de un *rover* durante la ejecución de su misión en un entorno planetario. Lo que se buscaba era encontrar nuevos métodos de localización utilizando sistemas de visión que pudiesen minimizar los errores de localización. La figura 1.2 muestra el robot utilizado por Lacroix et al. en su trabajo [50]. En él proponen una técnica de localización basada en un sistema de visión estéreo siguiendo el esquema de trabajo utilizado por Moravec y explicado anteriormente. Incorpora además el concepto de refinamiento en la localización. Mientras se ejecuta el proceso de localización basado en la odometría visual, se guarda información en 3-D de los objetos que se van encontrando en la escena, que denomina *aspectos*, así como de la distancia y orientación de la cámara respecto a dichos objetos. Por cada imagen se busca si existe alguno de esos objetos guardados en la imagen actual. En el caso de que se encuentre algún objeto ya conocido, se refinará la posición actual del robot. Las características que se guardan de cada uno de los *aspectos* son valores geométricos como el volumen, momento de inercia, etc o valores fotométricos como puede ser la textura. Una vez detectado un objeto, se compara con los objetos registrados con anterioridad y en caso de encontrar una correlación, se recalcula la posición actual en función de la posición del objeto guardado. En dicho trabajo, los objetos se limitaban a piedras en un entorno plano, esto es, se implementaba un segmentador de objetos 3-D sencillo que detectaba cambios en el relieve en un entorno plano.

Estos dos trabajos comentados, [67, 50], suponen dos hitos fundamentales en el campo de la odometría visual, aunque han dejado abiertas muchas cuestiones como se pondrá de manifiesto en la revisión del estado del arte en el Capítulo 2, y que sirven de motivación para la realización de esta tesis doctoral.

## 1.2. Motivación

Actualmente, la mayor parte de los algoritmos de odometría visual propuestos no trabajan en tiempo de ejecución, sino que los resultados se obtienen mayoritariamente *offline*, esto es, se toman las imágenes y posteriormente se realiza el procesamiento de las mismas. Otros trabajan en una frecuencia de toma de imágenes muy baja, lo que implica que se precisa un movimiento del robot muy lento. Esto es debido a que los



Figura 1.2: Robot utilizado en el trabajo de Lacroix et al.

algoritmos actuales de odometría visual conllevan cálculos muy pesados y de alta carga computacional. Esto implica que para su correcta ejecución, muchas veces es necesario disponer de equipos informáticos de alto nivel, con el coste que ello conlleva. Los sistemas de visión que se utilizan suelen ser sistemas de visión avanzados de alto coste.

El beneficio de utilizar una única cámara puede venir dado por diferentes factores, siendo los más importantes el factor económico y el factor de coste computacional al tratar información en 2-D en comparación al tratamiento de información tridimensional.

Por otra parte, el manejo de oclusiones de objetos y de movimientos en la escena y en el campo de visión no está resuelto y produce errores graves en el cálculo de la odometría visual. Los métodos existentes calculan en ocasiones la velocidad del vehículo basándose en otros sensores del robot móvil. Por último, la utilización de suelos texturizados o estructurados resulta requisito imprescindible para el buen funcionamiento de los métodos actuales de cálculo de odometría visual.

Muchos de los trabajos propuestos con anterioridad se basan en el cálculo del movimiento a partir de la estructura, esto es, el cálculo de la estructura en 3-D a partir de una secuencia de imágenes capturadas en diferentes instantes de tiempo y, con ello, el cálculo del movimiento realizado. En cambio, el trabajo que se presenta en esta Tesis basa la



estimación del movimiento que ha tenido lugar calculándolo como un giro sobre un punto denominado Centro Instantáneo de Rotación. Los trabajos previos necesitan en muchas ocasiones de sensores internos para calcular el movimiento realizado por el robot, ya sea para calcular la velocidad del robot, o para calcular los giros que han tenido lugar.

### 1.3. Objetivos

En la robótica móvil, la necesidad de conocer la localización precisa de un robot durante una navegación se torna en imprescindible para cualquier objetivo que se desee efectuar durante dicha navegación. El principal objetivo de esta tesis es afrontar el problema de la localización de un robot móvil en un entorno heterogéneo basándose única y exclusivamente en sus sensores odométricos, de forma que utilizando el sistema de localización aquí presentado se pueda calcular de forma correcta la localización del robot en entornos no ideales.

En esta tesis, se presenta un algoritmo de odometría visual puro, que es fiable, robusto y tolerante a fallos tras deslizamientos, bloqueos de ruedas, baches, etc. La solución aquí presentada proporciona un cálculo de odometría visual en tiempo real utilizando elementos computacionales de bajo coste y un sistema de visión de bajo coste. Los elementos utilizados se pueden calificar como domésticos. Este algoritmo de cálculo de odometría visual puede utilizarse como un sensor independiente sin la necesidad de añadir ningún otro sensor. Se presenta también como caso práctica de aplicación, la fusión sensorial con otros sensores, en este caso la fusión sensorial de odometría visual y odometría de las ruedas, de forma que se logra un sensor odométrico fiable y robusto en entornos heterogéneos y en diferentes contextos.

A partir de esta idea, los objetivos concretos de esta tesis son los siguientes:

- Desarrollar un procedimiento de odometría visual puramente exteroceptivo fiable y robusto que sea capaz de realizar el cálculo del movimiento de un robot a partir de las imágenes capturadas por una única cámara que se encuentre sobre el robot. Dicho algoritmo ha de ser viable para ejecutarse en tiempo real durante la navegación del robot haciendo uso de elementos computacionales

domésticos.

- Diseñar y desarrollar un sistema de localización basado en el filtro extendido de Kalman (EKF), e integrar conjuntamente el sistema de odometría visual con el sistema de odometría de las ruedas, logrando de esta manera un sistema odométrico más robusto y fiable en diferentes entornos.

## 1.4. Metodología

La metodología seguida durante el desarrollo de esta investigación sigue una estructura planeada previamente a la realización del trabajo y que se basa en los principios científicos necesarios para un trabajo de este tipo. Primeramente se realiza un estudio de la literatura relacionada existente, centrándose en el estudio de los diversos trabajos de localización basada en el cálculo de la odometría visual tanto utilizando una única cámara como de aquellos trabajos en los que se utilizan varias cámaras. Se estudian las ventajas e inconvenientes de cada una de las propuestas que se han realizado con anterioridad.

Posteriormente, se realiza un diseño del método de odometría visual que se va a desarrollar realizando de forma individualizada el estudio de cada uno de los componentes buscando una solución final óptima tanto en cuanto a resultados como en cuanto a tiempos de ejecución de cada ciclo.

Una vez que se dispone de un algoritmo de cálculo de odometría visual, se realiza la aplicación de una fusión sensorial con otros sensores odométricos de forma que se logre la generalización en cuanto a entornos de utilización y la robustez del sistema odométrico resultante de la fusión sensorial.

La metodología de investigación seguida durante el desarrollo de este trabajo se divide en las siguientes actividades:

1. Estudio de los conceptos básicos y trabajos relacionados
  - a) Métodos de localización basados en visión artificial
  - b) Métodos de odometría visual monocular (una única cámara)
  - c) Métodos de odometría visual estéreo (varias cámaras)

- ## 1.5. Estructura del documento

## PARTE I

La primera parte de este documento corresponde al capítulo actual, en el que se presenta una introducción a los sistemas de localización basados en odometría visual y de ruedas utilizados comúnmente en la robótica móvil. Se presentan también la motivación y objetivos de esta tesis. De la misma manera, se presenta también la metodología seguida durante el proceso científico.

## **PARTE II**

En la segunda parte de este documento se realiza la exposición resumida del estado del arte tanto en sistemas de localización basado en odometría visual como en sistemas de localización semejantes, ya sea por el uso de sensores odométricos como por la semejanza que se tienen en el cálculo de la posición actual de un robot móvil. Se expone también la evolución que ha tenido a lo largo del tiempo los métodos de cálculo de odometría visual tanto monocular como estéreo.

## **PARTE III**

En la tercera parte se expone con detalle el procedimiento en tiempo real que se propone en esta tesis para el cálculo de la odometría visual, utilizando una única cámara que se encuentra situada sobre el robot que realiza la navegación. Se explica paso por paso cada uno de los procesos que tienen lugar en la ejecución del cálculo de la odometría visual.

## **PARTE IV**

En la cuarta parte se expone una muestra de la batería de pruebas realizadas para la comprobación correcta tanto del procedimiento de odometría visual en diferentes entornos. Se realiza también un estudio sobre los tiempos de ejecución para mostrar la adecuación a la ejecución en tiempo real del procedimiento propuesto en esta tesis.

## **PARTE V**

En la quinta parte de este documento se explica el método de fusión sensorial utilizando un Filtro de Kalman Extendido. Se toma como entrada del sistema los datos de odometría de las ruedas del robot y como medida de observación, los datos de odometría visual calculados siguiendo el procedimiento propuesto. Se aportan también los resultados de esta fusión sensorial, logrando un sensor odométrico completo, fiable y robusto.

## **PARTE VI**

Por último se expone una parte de conclusiones, discusión de los resultados, evaluación sobre la obtención de los objetivos propuestos inicialmente. Se muestra también unas vías de posible continuación del trabajo realizado.

## Capítulo 2

# Estado del arte

En este apartado se va a realizar una revisión de los trabajos que han contribuido al desarrollo de la odometría visual en sus distintas vertientes. Con ello se pretende conocer el estado actual de las técnicas en odometría visual y así poder encuadrar nuestro trabajo de tesis.

Los procedimientos de cálculo de odometría visual se pueden clasificar primeramente en dos bloques claramente diferenciados: aquéllos que utilizan una única cámara y aquéllos que utilizan más de una cámara. En el caso de la odometría visual estéreo, es fácil obtener la estructura 3-D del entorno y, por tanto, la toma de medidas. En el caso de la odometría visual monocular se ha de desarrollar algún método para poder realizar mediciones, bien sea porque se conozca el tamaño de algún elemento concreto de la imagen, o porque existan restricciones de movimiento o porque se integre junto con algún otro sensor.

Esta clasificación ha servido para estructurar la revisión del estado del arte. Ha sido de gran ayuda para el autor la correcta clasificación y estructuración en las revisiones realizadas por D. Scaramuzza y F. Fraundorfer en [90, 23] de las diversas soluciones aportadas a lo largo de los últimos años.

## 2.1. Odometría visual estéreo

La odometría visual estéreo utiliza las imágenes capturadas a partir de más de una cámara o la simulación del uso de varias cámaras, como en el caso de Moravec [67], que utilizaba una única cámara pero tomaba varias imágenes en una misma localización del robot.

Los pasos a seguir durante la estimación del movimiento utilizando técnicas de odometría visual estéreo se encuadran en el esquema ya utilizado inicialmente por Moravec y explicado con anterioridad en el apartado 1.1: detección de características, cálculo de la estructura en 3-D del entorno -se realizará un modelo de error del cálculo de la triangulación estéreo-, seguimiento de los puntos característicos a lo largo de una secuencia de imágenes, eliminación de valores anómalos y estimación del movimiento.

Matthies y Shafer [61] formulan un nuevo método de modelado del error en la triangulación estéreo basado en una distribución gaussiana en tres dimensiones. Utilizando el procedimiento de Moravec con un sistema de visión binocular y sustituyendo el modelo de error escalar en la triangulación por un modelo basado en una distribución gaussiana, lograron un error del 2 % en un recorrido de 5,5 metros y un error de 1° en la orientación.

Olson et al. [76, 77], demostraron que el aumento en el orden del error era mayor que lineal debido principalmente a los fallos en el cálculo de la orientación. Un fallo durante el avance implica únicamente un error en la distancia recorrida propagado en el tiempo. Sin embargo, un error en el cálculo de un giro implica consecuencias mayores, ya que los próximos cálculos de distancia recorrida conllevarán también un error en la posición y la orientación. Este error será mayor que  $O(n)$  con el paso del tiempo, siendo  $n$  la distancia recorrida. Por ello, utilizaron un sensor de orientación absoluto que corregiría la orientación en los giros, disminuyendo así el error total. De esta manera el error que proporcionaban en sus resultados era del 1 % en un recorrido de 20m. La Figura 2.1a) [77] muestra que el error aumenta de forma lineal con la distancia cuando se tiene un sensor de orientación absoluto. Sin embargo cuando no se posee dicho sensor, el error es de orden superior a  $O(n)$ . La Figura 2.1b) muestra el recorrido completo. La línea continua refleja la posición del robot en cada instante de tiempo tomada con un sensor GPS, mientras que la línea punteada muestra la posición en cada instante de tiempo uti-

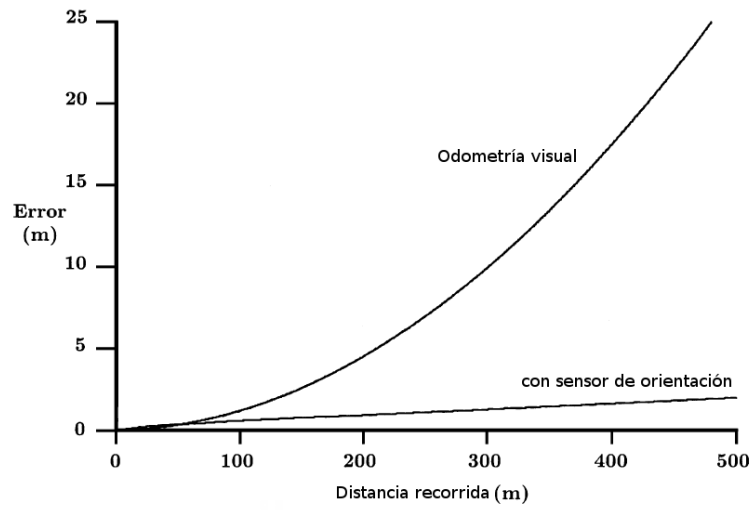
lizando el método propuesto. La línea discontinua segmenteada muestra el cálculo utilizando únicamente la odometría visual.

El trabajo de Lacroix et al. [50], explicado en el apartado 1.1, se encuadra dentro de las técnicas de odometría visual estéreo. Este trabajo sirve de base para otros trabajos realizados posteriormente [64, 41] que utilizan la misma técnica de detección y seguimiento de puntos.

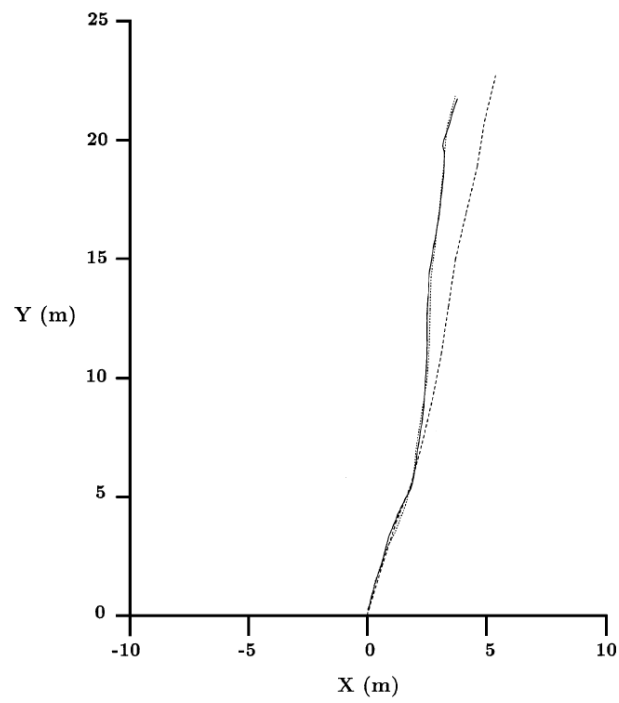
Cheng et al. [9] toman como base el trabajo de Lacroix en su implementación final de odometría visual sobre el *Mars rover*. Mejoraron la implementación inicial de Olson et al. [76, 77], en dos aspectos. Tras la detección de esquinas, mediante el detector de esquinas de Harris, utilizan el modelado del error propuesto por Lacroix et al. Utilizaron en el paso de estimación del movimiento el algoritmo RANSAC, *RAndom SAmple Consensus* [20] para la eliminación de valores anómalos, como proponía Nister [74]. Maimone et al. [59] mostraron los resultados del uso de la odometría visual en los diferentes *Mars rovers* enviados para la exploración de Marte. La figura 2.2 muestra los resultados de los trabajos de Cheng et al. [9, 59] en un entorno completamente adverso para la odometría de las ruedas como lo es durante el proceso de ascensión en un terreno rocoso.

Milella y Siegwart [64] proponen una solución diferente al problema de la estimación del movimiento y la eliminación de valores anómalos para un vehículo todo-terreno. Utilizan el esquema habitual hasta el momento, que se basa en tres fases principales: detección de características, seguimiento de las mismas y finalmente la estimación del movimiento. Primeramente, utilizan el motor estéreo SRI [47], con patente activa actualmente, y posteriormente el detector Shi-Tomasi [94] para encontrar las características sobre las que se realizará el seguimiento. Una primera manera de eliminar los posibles errores es tomar sólo aquellas características que tengan valores 3-D fiables y que tengan correspondencia mutua, esto es, que dadas dos imágenes en dos instantes de tiempo,  $I_1$  e  $I_2$ , una característica  $p_1$  (en la primera imagen),  $p_2$  en la segunda imagen es fiable si  $p_1$  es el punto con mayor coeficiente de correlación para  $p_2$  y, también,  $p_2$  es el punto con mayor coeficiente de correlación respecto de  $p_1$ . Posteriormente, se realiza un filtro estadístico de valores anómalos basado en la mediana y la desviación típica [26]. La estimación del movimiento se realiza utilizando el algoritmo ICP, *Iterative Closest Point*. En el proceso final lo que se busca es encontrar la matriz de transformación 3-D,  $T$ , que minimiza la función objetivo:





a)



b)

Figura 2.1: Resultados del trabajo de Olson et al.

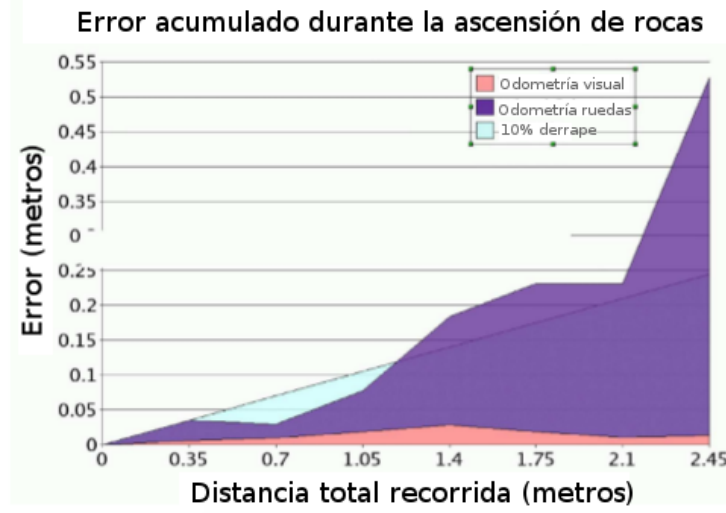


Figura 2.2: Resultados del trabajo de Cheng et al.

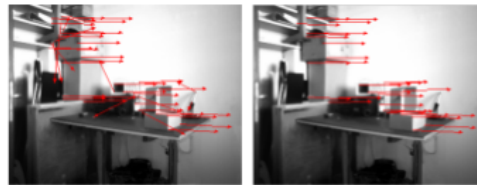
$$F(T) = \frac{1}{N} \sum_{i=1}^N \|TP_2^i - P_1^i\|^2 \quad (2.1)$$

donde  $P_1^i$  y  $P_2^i$  son los puntos en 3-D en dos instantes de tiempo sucesivos y  $N$  es el número de parejas de puntos de los que se dispone. La figura 2.3a) muestra los puntos característicos detectados en dos imágenes consecutivas (imagen izquierda en la cámara estéreo). La figura 2.3b) muestra el seguimiento de características inicial -figura de la izquierda- y el resultado tras la eliminación de características 3-D no fiables y tras el filtrado basado en la mediana -figura de la derecha-. Por último, la figura 2.3c) muestra el robot utilizado por los autores durante la ejecución de las pruebas.

Nister et al. [74] propusieron una solución, estéreo y también otra monocular, completamente diferente a las anteriores. Este artículo se conoce no sólo por acuñar el término odometría visual, sino por proporcionar la primera implementación en tiempo real. Nister et al. mejoraron las implementaciones anteriores en diversos aspectos. Primeramente, no realizaron el seguimiento de los puntos característicos entre imágenes, sino que detectaban las características, utilizando el detector de Harris, independientemente en cada imagen capturada y buscaban posteriormente concordancias entre las características. La optimización lograda consistía



a)



b)



c)

Figura 2.3: Imágenes relativas al trabajo de Milella y Siegwart. a)Puntos característicos detectados en dos imágenes consecutivas. b)Seguimiento de las características y filtrado de las mismas. c) Robot utilizado

en detectar características en las imágenes tomadas en dos instantes de tiempo diferentes de forma independiente. Por cada característica de la primera imagen, se buscaba una concordancia sólo con los puntos característicos de la segunda imagen que estuvieran a una distancia inferior a un umbral, y no con todos los puntos de la segunda imagen. Para que una concordancia entre dos características fuese dada por válida, la consistencia mutua, ya explicada con anterioridad, se debía cumplir. Una vez realizado el proceso de detección y seguimiento de características, la estimación del movimiento según su propuesta estéreo, se realiza utilizando un conjunto de imágenes -y sus respectivas características- como entrada del RANSCAC preventivo [73] de 3 puntos.

Un esquema diferente para la estimación del movimiento fue introducida por Comport et al. [14]. En vez de utilizar técnicas de registro 3-D a 3-D o una estimación de la posición de la cámara 3-D a 2-D, se basaron en el tensor cuadrifocal, *Quadrifocal tensor*, que permite calcular el movimiento a partir de las coincidencias en la imagen 2-D a 2-D, sin tener que triangular los puntos en ninguna de las parejas de imágenes estéreo. El beneficio de esto es un cálculo de la estimación del movimiento mucho más preciso. La figura 2.4 muestra la base del cálculo de las correspondencias a partir de los cuatro puntos posibles: punto inicial,  $p$ , y final,  $p''$ , de la cámara de la izquierda y punto inicial,  $p'$ , y final,  $p'''$ , en la cámara de la derecha.

## 2.2. Odometría visual monocular

En la odometría visual monocular tanto la estructura 3-D del entorno como el movimiento relativo deben ser calculados a partir únicamente de los datos en dos dimensiones. La medición de las distancias en imágenes 2-D no es posible de forma absoluta, por lo que se establece una distancia relativa al primer movimiento. Es decir, se toma como unitario -o como referencia- el movimiento que ha ocurrido en el primer instante y a partir de él se calculan los movimientos sucesivos [35]. A lo largo de los últimos años se han realizado trabajos con resultados satisfactorios en largas distancias utilizando cámaras comunes u omnidireccionales [75, 15, 51, 28, 105, 65, 92, 8, 86].

Los trabajos de odometría visual se pueden clasificar en tres categorías: basados en características, globales e híbridos. Los métodos basados

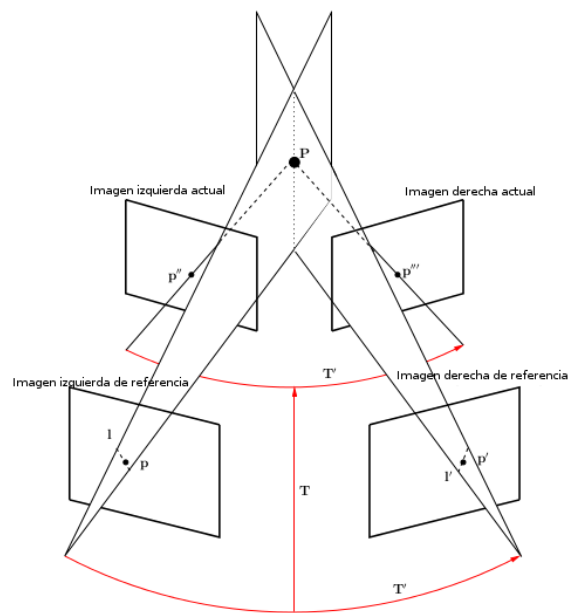


Figura 2.4: Comport: tensor cuadrifocal

en características utilizan puntos de la imagen que son fácilmente localizados y repetibles a lo largo de la secuencia de imágenes. Los métodos globales utilizan la información de intensidad de todos los píxeles de la imagen o bien de una región de la misma. En cuanto a los métodos híbridos, utilizan una mezcla de ambos métodos.

### 2.2.1. Métodos basados en características

Siguiendo la clasificación anterior, los trabajos [74, 15, 51, 105, 69, 91, 82] se encuadran dentro de esta primera categoría. La primera propuesta en tiempo real con una única cámara fue presentada por Nister et al. [74]. Para el paso de la información en 2-D a los cálculos realizados en 3-D utilizan hipótesis geométricas. Para el cálculo de la estimación de movimiento utilizan un solucionador de cinco puntos [72] dentro de RANSAC. A partir de este trabajo, el RANSAC de cinco puntos fue el método más común a la hora de abordar los problemas de odometría visual y flujo óptico.

Lhuillier [51] y Mouragnon et al. [69] presentaron una solución basada en la utilización de una ventana temporal de imágenes a partir de la que se reconstruía el movimiento y se realizaba un mapa 3-D. La ventana permitía tomar como entrada las  $m$  últimas imágenes de la secuencia. Utilizaron el RANSAC de cinco puntos [72] para la eliminación de valores anómalos.

Tardif et al. [105] presentan una solución semejante a las aportadas hasta el momento con el añadido de la creación de un mapa 3-D. El esquema del procedimiento es similar a los propuestos con anterioridad: detección de características (utilizando SIFT [57]), seguimiento de características, cálculo de la posición y orientación de la cámara y eliminación de valores anómalos utilizando una versión optimizada del RANSAC de cinco puntos (RANSAC preventivo) [73]. El cálculo del movimiento se realiza únicamente utilizando la imagen del muestreo previo, esto es, no realiza ninguna optimización con imágenes de muestreos anteriores. A cambio, realiza un cálculo de la estructura del entorno, creando un mapa en 3-D de lo recorrido. La figura 2.5 muestra la creación de dicho mapa durante el recorrido presentado de ejemplo en el artículo. El recorrido tiene una longitud de 2,5 km.



Figura 2.5: Creación del mapa en 3-D por Tardif

### 2.2.2. Métodos globales

Los trabajos [28, 65, 92] se encuadran dentro de la categoría de soluciones globales, que utilizan la información de intensidad de todos los puntos de la imagen. Milford y Wyeth [65] presentaron un método para extraer la velocidad de rotación y de avance a partir de una cámara montada en un vehículo, que después fue utilizada en un esquema RatS-LAM [66]. Utilizaron un seguimiento de un patrón que se encuentra en el centro de la escena.

### 2.2.3. Métodos híbridos

El mayor problema con los procedimientos globales es que no son robustos en caso de que aparezcan oclusiones de objetos en las imágenes. Es por ello que Scaramuzza y Siegwart [92] utilizaron un procedimiento global para el cálculo de la rotación de un vehículo y un procedimiento basado en características para estimar el avance del mismo. El procedimiento basado en características se utiliza también para encontrar fallos en el procedimiento global. En resumen, se tenían dos *trackers* diferentes. El primero de ellos estaba basado en la homografía, y detecta y realiza el seguimiento de características ubicadas en el suelo. El segundo de ellos se basa en el seguimiento de regiones y es el encargado de calcular la rotación que se produce. Introducen el concepto de brújula visual luego utilizada posteriormente en diversos trabajos. La figura 2.6 compara los resultados del cálculo de la odometría visual utilizando un método híbrido (global y basado en características) respecto a utilizar únicamente el método basado en características.

## 2.3. Optimización en el cálculo de la odometría visual

Todos los métodos anteriores fueron diseñados para calcular la odometría visual en vehículos sin restricciones de movimiento en los seis grados de libertad. Sin embargo, existen otros métodos de cálculo de odometría visual que han sido diseñados específicamente para algún tipo de vehículo con restricciones de movimiento conocidas. De esta manera logran un menor tiempo de cálculo y una mayor precisión en el cálculo.



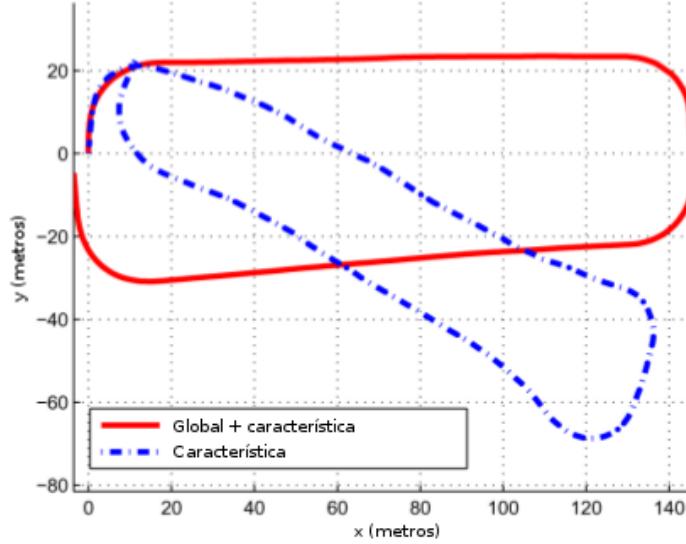


Figura 2.6: Comparación de cálculo de odometría visual utilizando un método híbrido (global y basado en características) y un método basado únicamente en características, por Scaramuzza

lo del movimiento. Por ejemplo, Liang y Pears [52], Ke y Kanade [45], Wang et al. [111] y Guerrero et al. [32] utilizaron homografías para estimar el movimiento. Scaramuzza et al. [91, 87] introdujeron un RANSAC de un único punto gracias a considerar de las restricciones cinemáticas del vehículo y asumiendo un movimiento planar. Las restricciones de un vehículo no holónomo han sido utilizadas por Pretto et al. [82] para mejorar el seguimiento de características y por Fraundorfer et al. [24] para la determinación de la ventana temporal.

Como se ha comentado, el cálculo de la posición a partir de la odometría visual entre dos imágenes introduce un error en la localización en cada instante de tiempo. De esta manera, la posición final calculada contiene un error que aumenta con el paso del tiempo. El principal objetivo de la odometría visual es calcular las transformaciones relativas que ocurren entre dos imágenes  $I_k$  e  $I_{k-1}$ . Una vez calculadas, se concatenan todas las transformaciones para obtener la trayectoria que ha tenido lugar. La odometría visual reconstruye dicha trayectoria paso a paso. Inevitablemente se introduce un error en cada paso. Se han realizado trabajos para tratar de reducir ese error que se va acumulando paso

a paso. Algunos trabajos [24, 101, 48, 104] han introducido una ventana temporal y local de  $m$  imágenes (períodos de muestreo).

Ese error puede reducirse también utilizando otros sensores (como se pretende en nuestro trabajo), como puede ser un GPS, un láser o un IMU [48, 70, 43]. Por ejemplo, Konolige et al. [48] realizan el cálculo de la odometría visual siguiendo el esquema tradicional: extracción de características, seguimiento de las mismas (utilizando ZNCC, *Zero-mean Normalized Cross Correlation*) y estimación del movimiento utilizando RANSAC de 3 puntos. Posteriormente añaden una optimización de dicho movimiento utilizando los últimos  $m$  movimientos. Por último, realizan una fusión sensorial con un sensor inercial utilizando el filtro de Kalman extendido. Presentan los resultados tras realizar un recorrido de unos 10 km y demuestran que son capaces de reducir el error en un factor comprendido entre 2 y 5.

## 2.4. Cálculo del movimiento

El cálculo del movimiento es el paso principal que se tiene en un procedimiento de cálculo de odometría visual. Es en este paso en el que se calcula el movimiento que ha realizado la cámara entre la imagen capturada en un instante de tiempo  $k$  y la imagen capturada en el instante anterior,  $k - 1$ . Concatenando todos estos movimientos se puede reconstruir la trayectoria total que ha realizado la cámara y por tanto la del vehículo o robot en el que se encuentra. Para calcular la transformación ocurrida entre dos imágenes  $I_k$  e  $I_{k-1}$  se parte de dos conjuntos de características  $f_k$  y  $f_{k-1}$  que se obtienen de los instantes de tiempo  $k$  y  $k - 1$  respectivamente. Estas características pueden estar definidas en dos o en tres dimensiones. En función de ello, existen diferentes métodos: 2-D a 2-D, en este caso ambos conjuntos están definidos en dos dimensiones; 3-D a 3-D, ambos están definidos en tres dimensiones; 3-D a 2-D, en este caso  $f_{k-1}$  está definido en tres dimensiones y  $f_k$  son las proyecciones en la imagen (2-D) de dichos puntos.

Las características en una imagen pueden ser puntos o líneas. En general, dado que la búsqueda de líneas en imágenes reales es compleja, en odometría visual, las características utilizadas suelen ser puntos. En [42] se presenta una revisión en profundidad de los usos de características (líneas y puntos).

### 2.4.1. Detección y seguimiento de características

En cuanto a los métodos de detección de puntos característicos en una imagen y el posterior seguimiento de los mismos, se puede hablar de dos estrategias diferentes. Por un lado se pueden localizar los puntos característicos en la imagen inicial y posteriormente realizar la búsqueda de dichos puntos en las sucesivas imágenes, o bien se pueden detectar de forma independiente los puntos característicos en cada imagen y posteriormente encontrar coincidencias de puntos en las diferentes imágenes. La primera opción es más lógica cuando los movimientos son pequeños. La segunda opción se utiliza cuando los movimientos son más grandes o cuando cambia el punto de visión. En los primeros trabajos de odometría visual [67, 61, 50, 76] es más común encontrar la estrategia de detección y posterior seguimiento de características, mientras que en los trabajos de la última década [72, 51, 69, 105, 91], es más común la utilización de detectores de características de forma independiente por cada imagen. Esto es debido a que el tipo de problema estudiado ha cambiado. Inicialmente, se trataba con movimientos pequeños y con la cámara montada cerca de la escena de estudio. Sin embargo, en los problemas actuales se trata con movimientos mayores o con escenas más amplias.

Durante el proceso de detección de características se busca detectar puntos que sean fácilmente distinguibles y comparables en una imagen. Los detectores de características se dividen en detectores de esquinas y detectores de *blobs*. Una esquina se define como un punto que se sitúa en la intersección de dos o más bordes. Un *blob* es una región de la imagen que claramente se diferencia de su entorno en términos de intensidad, color y textura. No es una esquina ni tampoco un borde.

Las propiedades de un buen detector de características en una imagen son: precisión en la posición dentro de la imagen, repetibilidad (una característica existente en dos imágenes debería ser detectada en ambas imágenes), eficiencia en tiempos de ejecución, robustez (ante ruido, imagen desenfocada), distinción (de manera que las características sean claramente identificables entre ellas), variación nula (ante cambios en la luminosidad y ante cambios geométricos como puede ser rotación, escalado (zoom), distorsiones de perspectiva) [110, 2, 5].

Los detectores de esquinas más utilizados en trabajos de odometría visual son los de Moravec [67], Forstner [21], Harris [34], Shi-Tomasi [94], y FAST [85]. Por otra parte, entre los detectores de *blobs* utilizados des-

tacan SIFT [56], SURF [3], y CENSURE [1]. En [95] y [110] podemos encontrar una revisión completa de los detectores de características en una imagen. Cada detector tiene sus pros y sus contras. Los detectores de esquinas son más rápidos pero son menos diferenciadores, lo que puede llevar a errores en el seguimiento de características. Por otra parte, los detectores de *blobs* son más diferenciadores pero más lentos. Además, los detectores de esquinas permiten una mejor localización de la característica dentro de la imagen, pero sin embargo son menos robustos ante cambios de escala. Esto significa que las esquinas no se detectan con tanta facilidad como los *blobs* en cambios de escala (zoom) o en cambios del punto de visión.

Por ello, la elección del detector de características que se va a utilizar no es trivial, sino que debe ser un proceso en el que se evalúen las necesidades del trabajo a realizar. Se han de tener en cuenta las restricciones computacionales (de recursos y temporales), tipo de entorno y tipo de movimiento. En próximos apartados se explicará con más detalle el proceso de elección del detector de características utilizado durante este trabajo y las pruebas realizadas para demostrar la toma de la decisión. La Figura 2.7 [23], muestra una comparativa de las propiedades de los detectores más comunes en la literatura. Una evaluación sobre el rendimiento de los detectores de características utilizados en el cálculo de odometría visual en interiores puede verse en [93] y para exteriores en [91] y [93]. El detector de Shi-Tomasi es un detector de esquinas que es invariante en rotaciones pero no en cambios de escala, y tiene altos niveles de repetibilidad y mucha precisión en la detección del punto característico. Es además un detector de bajo consumo computacional y por tanto favorece su implementación en un sistema bajo restricciones de tiempo real.

Una vez que se han localizado los puntos característicos en una imagen, el siguiente paso es lograr identificar inequívocamente esa característica en las siguientes imágenes. Para ello, se ha de generar un descriptor de esa característica. Para la creación de este descriptor se toma la región que se encuentra alrededor del punto característico, de forma que a la hora de buscar similitudes en las siguientes imágenes, se comparará el píxel del punto característico, junto con los píxeles vecinos. El descriptor más simple sería aquél que toma como valores identificables los valores de intensidad del punto característico y de los píxeles vecinos. De esta manera, la comparación de descriptores se hace con funciones de error

	Corner Detector	Blob Detector	Rotation Invariant	Scale Invariant	Affine Invariant	Repeatability	Localization Accuracy	Robustness	Efficiency
Harris	x		x			+++	+++	++	++
Shi-Tomasi	x		x			+++	+++	++	++
FAST	x		x	x		++	++	++	++++
SIFT		x	x	x	x	+++	++	+++	+
SURF		x	x	x	x	+++	++	++	++
CENSURE		x	x	x	x	+++	++	+++	+++

Figura 2.7: Comparativa de los detectores de características: propiedades y rendimiento, por Scaramuzza

como el sumatorio de la diferencia de cuadrados (SSD) de intensidades o la correlación cruzada normalizada (NCC) [29]. NCC funciona mejor que SSD en cambios de luminosidad y brillo. Otra medida de similitud más robusta es la transformada de Census [113], que convierte cada área de la imagen en un vector binario que representa qué píxeles vecinos tienen la intensidad superior o inferior a la intensidad del punto central. La similitud entre áreas se calcula utilizando una distancia de Hamming.

Este tipo de descriptores tienen muchas limitaciones. Los descriptores pueden ser distintos con los cambios de orientación, escala, punto de vista de la cámara. De hecho ni NCC ni SSD son invariantes ante estos cambios. Por ello, estos descriptores se suelen utilizar únicamente para imágenes donde el punto de visión es muy próximo a los objetos de la imagen. Uno de los descriptores más utilizados es el descriptor SIFT. Consiste en un histograma de los gradientes de orientación locales. El área alrededor del punto característico se divide en  $4 \times 4$  celdas. Para cada cuadrante, se construye un histograma de 8 gradientes de orientación. La Figura 2.8 muestra el proceso de cálculo de este descriptor en un punto de la imagen.

De esta manera, el descriptor SIFT demuestra ser estable ante cambios en la iluminación, la rotación, el escalado e incluso a cambios en el punto de visión, hasta  $60^\circ$ . El descriptor SIFT funciona mejor en *blobs* que en esquinas. Esto es debido a que las esquinas, por definición, se encuentran en las intersecciones de los bordes y los *blobs* se encuentran en regiones altamente texturizables.

Para realizar el seguimiento de las características a lo largo de la secuencia de imágenes existen diversas opciones. Si se plantea como un

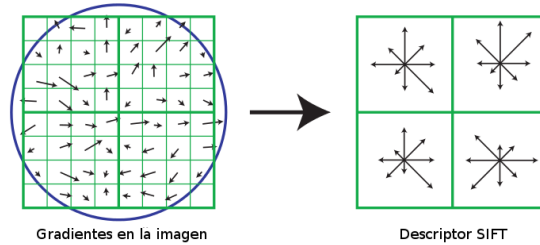


Figura 2.8: Ejemplo de una característica y el descriptor SIFT

problema de coincidencia (*matching*) de descriptores, lo que se ha de hacer es comparar todos los descriptores de la primera imagen con todos los descriptores de la segunda imagen. Esta comparación se realizará utilizando algún tipo de medida de similitud. Por ejemplo, si el descriptor es la luminosidad en una ventana alrededor de una característica, se puede utilizar SSD o NCC como medida de similitud. Para los descriptores SIFT, la distancia euclídea es una buena medida.

Tras comparar todos los descriptores de características de la primera imagen con todos los descriptores de características de la segunda imagen, se escoge (para cada descriptor de la primera imagen) el mejor descriptor de la segunda imagen (en términos de mayor similitud o de menor distancia). Puede ocurrir que un mismo descriptor de la segunda imagen quede vinculado a más de un descriptor de la primera imagen. Para evitar este problema se puede realizar la comparación inversa, esto es, un descriptor de la primera imagen queda vinculado con un descriptor de la segunda imagen si y sólo si ambos descriptores son los descriptores más similares en la imagen el uno para el otro, y viceversa. Esto se denomina la comprobación de consistencia mutua. El problema de esta comprobación es el aumento del coste computacional que tiene lugar. Esta comprobación es difícilmente compatible con problemas con restricciones de tiempo real.

Por otro lado, se pueden realizar búsquedas de características en la segunda imagen sólo en la región de la imagen donde se espera que vayan a aparecer. Es decir, se puede realizar un modelo predictivo del movimiento de la cámara y, por lo tanto, reducir el espacio de búsqueda a una región en concreto. Este modelo puede ser generado a partir de la información obtenida de otros sensores, como por ejemplo un sensor inercial, odometría de las ruedas [59], sensor láser, un GPS, etc. Esta predicción

del movimiento también se puede calcular a partir de los movimientos anteriores, como se propone en [16].

Una alternativa a buscar características en todas las imágenes y, posteriormente, compararlas todas con todas, es buscar sólo características en la primera imagen y, posteriormente, buscar dichas características en las siguientes imágenes. Esta propuesta, que se basa en una búsqueda y posteriormente un seguimiento, encaja perfectamente con el problema de la odometría visual cuando los movimientos que se realizan entre dos imágenes son pequeños. Para estos casos, SSD y NCC funcionan correctamente. Sin embargo, si el seguimiento se realiza a lo largo de una secuencia de imágenes muy grande, puede ocurrir que entre la primera imagen de esa secuencia y la última imagen, las características sí hayan sufrido un cambio grande. El sistema de seguimiento utilizado en estos casos se denomina el *KanadeLucasTomasi (KLT) tracker* [94].

Las características KLT son aquellos puntos donde el valor propio de la matriz de gradientes de intensidad  $2 \times 2$  es mínimo. El seguimiento de las características se realiza utilizando el método Newton-Raphson minimizando la diferencia entre la ventana alrededor del punto característico en la primera imagen y la ventana alrededor de la característica en la segunda imagen.

Existen diversas implementaciones que optimizan el algoritmo de seguimiento de Lucas-Kanade [58]. Es el caso del trabajo de Bouguet [7] que realiza una implementación iterativa y piramidal del algoritmo de Lucas-Kanade. Esta estrategia consiste en procesar la imagen inicialmente en una menor resolución e ir refinando los resultados según se amplía la resolución de la imagen. De esta forma gestiona mejor las características que se encuentran en los bordes de la imagen, solventa el problema de la pérdida de coherencia espacial en movimiento grandes y es más eficiente desde el punto de vista computacional hablando (cerca de un 30 % más rápido).

Según estudios previos, como el realizado por Scaramuzza et al. [91], queda demostrado el peor funcionamiento en problemas de odometría visual del detector SIFT frente a otros sistemas como KLT. No es debido tanto a una pérdida de calidad en cuanto a las características o el seguimiento de las mismas, sino al número y distribución de las mismas. En imágenes con texturas uniformes, como las que podemos encontrar a lo largo de una navegación en entornos no cambiantes, el número de

características SIFT encontrados puede ser muy reducido y, por tanto, la distribución de los puntos significativos en la imagen puede no ser la adecuada. En el mismo trabajo se presentan otros resultados utilizando otros detectores como el de Harris [34] que presenta en esta ocasión una concentración de puntos característicos en algunas zonas de la imagen, así como una ausencia de los mismos en otras zonas de la imagen.

Esta distribución incorrecta y, en situaciones poco texturizadas incluso escasa, de los puntos característicos en la imagen conlleva errores futuros a la hora de realizar el seguimiento y el cálculo del movimiento que se ha producido entre dos imágenes.

Una vez planteada la base para la detección de características y el seguimiento de las mismas (o la búsqueda de características de forma independiente por cada imagen), se incorporan nuevos problemas y nuevas decisiones. La primera decisión que se ha de tomar será el número de características por cada imagen y la distribución de las mismas dentro de la imagen. Se ha demostrado que la distribución de las características en la imagen afecta de forma sustancial a los resultados de la odometría visual [74, 91, 98]). Como es obvio, un mayor número de características proporciona una mejor estimación del movimiento, pero del mismo modo, los puntos característicos han de cubrir la imagen de la mejor forma posible. Para hacer esto, la imagen se divide en celdas y el detector de características se aplica en cada una de las celdas.

Por otra parte, e influyendo en la fase de detección de características, existen alternativas a los detectores de características discretos. Se pueden utilizar métodos densos, como el flujo óptico [39], o métodos que no toman puntos característicos [60]. El objetivo del flujo óptico es el seguimiento individual de cada uno de los puntos de la imagen o de un conjunto de la imagen (utilizando alguna región de interés dentro de la imagen). Sin embargo, el flujo óptico asume también movimientos pequeños entre las imágenes. Otra alternativa sería utilizar métodos que no traten con puntos característicos, como el trabajo presentado en [60]. En este trabajo se presenta un método que utiliza todos los píxeles de ambas imágenes para calcular el movimiento relativo basándose en una transformación de Fourier armónica. Este método tiene como ventaja principal su correcto funcionamiento en imágenes con pocas texturas, aunque es computacionalmente muy costoso. La estimación del movimiento realizado es menos precisa que aquella que se realiza utilizando métodos con características.



### 2.4.2. Eliminar valores anómalos

El conjunto de puntos de los que se encuentran correspondencias en ambas imágenes puede contener valores anómalos, que vienen dados por errores en la asociación. Las posibles causas de esos errores pueden ser la aparición de ruido en las imágenes, oclusiones, objetos desenfocados, cambios en el punto de visión e iluminación, etc. Para que la estimación del movimiento realizado sea precisa, los valores anómalos han de ser eliminados. Se pueden utilizar métodos de estimación robustos como la estimación-M [107], eliminación de casos o eliminación directa de valores anómalos [96]. Estos métodos son suficientes en problemas donde la imagen es muy texturizada y por tanto los detectores de características funcionan mejor. Sin embargo, la solución más utilizada para la eliminación de valores anómalos está basada en las restricciones geométricas que introduce el modelo del movimiento. RANSAC [20] se ha convertido en el método estándar para la estimación del modelo en presencia de valores anómalos.

La idea en la que se basa RANSAC es obtener un modelo como hipótesis a partir de un conjunto de puntos y posteriormente verificar dicho modelo con otros puntos de la imagen. La hipótesis que consiga un mayor consenso con el resto de datos resultará la solución cierta. El número de subconjuntos que se han de utilizar (iteraciones del algoritmo) influye en la precisión del resultado final. RANSAC es un método probabilístico y no determinista. Esto significa que la solución que proporciona no es la misma en todas las ejecuciones. Sin embargo, cuanto mayor es el número de iteraciones, más estable será el resultado final.

De aquí se plantean dos vías de investigación para mejorar el rendimiento y eficiencia del algoritmo: minimizar el número de puntos necesarios y minimizar el número de iteraciones.

Si la cámara está calibrada, se puede calcular el modelo del movimiento con seis grados de libertad de la cámara (o del vehículo sobre la que se sitúa) utilizando un mínimo de cinco puntos característicos por cada imagen. La primera solución a este problema fue dada en 1913 por Kruppa [49]. Varios solucionadores de cinco-puntos fueron propuestos posteriormente [17, 80, 109], pero la implementación verdaderamente eficiente, basada en [80], fue proporcionada por Nister [72] en el 2003 y mejorada en [97]. Antes de eso, los solucionadores de seis [81], siete [100] y ocho puntos eran los que más se utilizaban. Por otra parte, también

existen solucionadores de  $n$ -puntos para cámaras omnidireccionales no calibradas [27, 103, 99], donde  $n$  depende del tipo de espejo o del ojo de pez utilizado. Lim et al. [53] demostraron que para cámaras omnidireccionales calibradas, el movimiento con 6 grados de libertad puede ser calculado utilizando sólo dos parejas de puntos de la imagen antipodales.

Durante los últimos años, se han hecho intentos de reducir el número de puntos necesarios para calcular el movimiento con 6 grados de libertad. En [25], Fraundorfer et al. proponen un solucionador basado en tres puntos para el caso de dos cámaras con ángulos de orientación conocidos. Este solucionador se puede utilizar cuando la cámara está ligada a un sensor de gravedad. Por otra parte, Naroditsky et al. [71] incidieron en ese trabajo al demostrar que el solucionador de tres puntos puede ser utilizado en un esquema RANSAC de cuatro puntos. Ese punto añadido, normalmente tomado como un punto en el infinito, se utiliza para fijar los dos ángulos de orientación de la cámara. Un solucionador de dos puntos fue propuesto por Kneip et al. [46] que utiliza una matriz de rotación completa a partir de los datos de un IMU que está rígidamente unido a la cámara.

En el caso de un movimiento sobre el mismo plano, los movimientos tienen tan sólo 3 grados de libertad. El modelo de movimiento en este caso puede ser calculado utilizando dos puntos como se describe en [78]. Para vehículos con ruedas, Scaramuzza et al. [91, 87] muestran que el movimiento puede ser descrito localmente como planar y circular, por lo que realmente los grados de libertad del movimiento se reducen a dos. Esto nos lleva a que el solucionador puede ser de un único punto. La utilización de un único punto se convierte en la parametrización mínima posible y por tanto del algoritmo RANSAC más eficiente. Una evaluación del funcionamiento de los RANSAC de cinco, tres y un puntos se presenta en [88]. En definitiva, la reducción de los puntos necesarios en RANSAC van ligados a un conocimiento previo, o a partir de otros sensores, del modelo de movimiento del vehículo sobre el que se encuentra la cámara.

Dado que el número de iteraciones no es fijo ni definible, pues depende del número de valores anómalos y de la probabilidad de éxito, hay muchos trabajos que se han centrado en aumentar la velocidad de ejecución de RANSAC. En [108] se logra que la medida de las correspondencias sea más fiable y se mejora la estimación de la hipótesis. [11] realiza una clasificación de las correspondencias basado en su similitud y genera las hipótesis de movimiento en base a las correspondencias mejor

posicionadas en dicha clasificación. El RANSAC preventivo [73] utiliza una puntuación preventiva para cada hipótesis y un número fijo de iteraciones. [83] incorpora el concepto de incertidumbre en los puntos característicos y muestra que esto determina una reducción en el número de potenciales valores anómalos y por tanto una reducción en el número de iteraciones. [63] propone un RANSAC determinístico que estima la probabilidad de que una correspondencia entre características sea cierta.

Todos estos métodos anteriores basan las hipótesis en función de los puntos característicos y sus correspondencias. Otros algoritmos operan tomando las hipótesis en función de los modelos de movimiento de un vehículo [13, 89]. El RANSAC preventivo es el más popular debido a que el número de iteraciones puede ser fijado a priori, lo que acaba teniendo muchas ventajas cuando se ejecuta en tiempo real, pues se tiene una certidumbre del coste temporal.

Sin embargo, y pese a que la tendencia más común en los trabajos realizados hasta el momento es la utilización del algoritmo RANSAC para el cálculo del movimiento que ha tenido lugar y la eliminación de valores anómalos, el coste computacional del algoritmo RANSAC ha llevado a que muchos de estos sistemas no sean capaces de trabajar en problemas de navegación en tiempo real. Para adecuar las condiciones de tiempo real a estas soluciones se han debido incorporar nuevos sensores o restricciones de movimiento para poseer más información y, por tanto, reducir el número de iteraciones del algoritmo.

En esa línea se encuentran trabajos como el de Scaramuzza et al. [91] que reduce el número de iteraciones necesarias en el algoritmo RANSAC a la hora del cálculo del movimiento incluyendo restricciones basándose en la cinemática del vehículo y asumiendo un movimiento planar. De esta manera, le basta realizar el seguimiento sobre un único punto. El número de iteraciones viene dado por la ecuación

$$N = \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^S)} \quad (2.2)$$

donde  $N$  es el número de iteraciones,  $S$  es el número mínimo de puntos,  $\epsilon$  es el porcentaje de valores anómalos esperados, y  $p$  es la probabilidad de éxito requerida. De esta forma, el número de iteraciones necesarios depende de forma exponencial respecto al número mínimo de puntos, por lo que si se logra reducir el número de puntos necesarios (en este

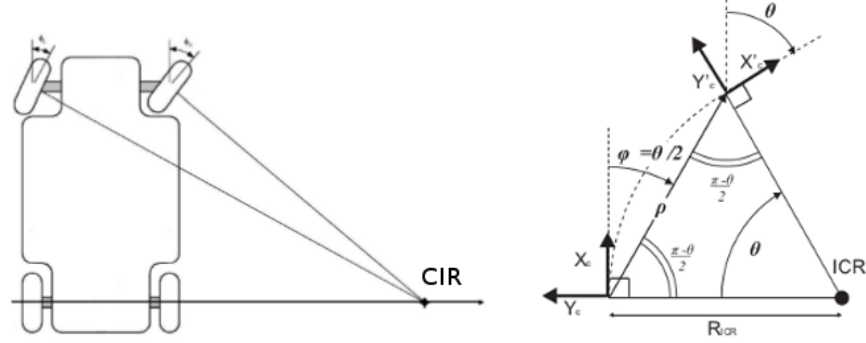


Figura 2.9: Scaramuzza: cálculo del Centro Instantáneo de Rotación en un giro

caso basándose en la cinemática del vehículo y asumiendo un movimiento planar), el número de iteraciones y, por tanto, el coste computacional del algoritmo se reduce exponencialmente.

Otra diferencia del trabajo presentado por Scaramuzza et al. respecto a los trabajos previos es que la estimación del movimiento se realiza basándose también en el cálculo del centro instantáneo de rotación, *CIR*, y no sólo en el cálculo de la estructura 3-D del entorno a partir del movimiento.

Se considera todo movimiento como una curva entorno a un punto fijo. Incluso un avance en línea recta se considera un giro entorno a un punto situado en el infinito. Por tanto, todo movimiento puede estar definido por el centro de rotación y por el ángulo girado. El centro de rotación se obtiene asumiendo que se conoce la cinemática del vehículo (geometría Ackermann) y la posición de las ruedas. Esto se muestra en la Figura 2.9.

Una vez que se conoce el centro de rotación, se ha de calcular el ángulo girado y de esta manera queda definido el movimiento que ha tenido lugar. Según queda demostrado en el artículo citado, el valor del ángulo girado,  $\theta$ , queda definido por la ecuación:

$$\theta = -2 \tan^{-1} \left( \frac{y'z - z'y}{x'z + z'x} \right) \quad (2.3)$$

siendo  $(x, y, z)$  el punto inicial, y  $(x', y', z')$  el punto final. Por tanto, tan

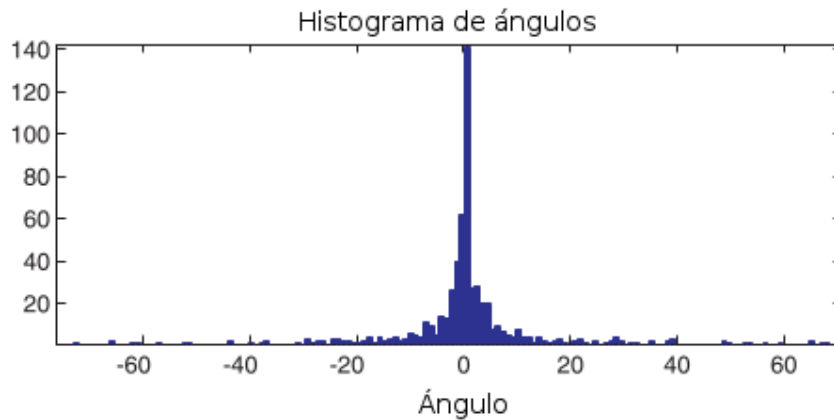


Figura 2.10: Scaramuzza: votación en el histograma de ángulos

sólo se necesita conocer un único punto para calcular el movimiento.

Aprovechando la reducción del coste temporal del algoritmo RANSAC, se realiza la ejecución de éste para varios puntos característicos y por tanto se obtienen varios resultados para una misma pareja de imágenes. Posteriormente se realiza una eliminación de valores anómalos de los ángulos de giro calculados utilizando un sistema de votación en un histograma, como se observa en la Figura 2.10.

Sin embargo, el sistema propuesto no es un sistema de odometría visual puro puesto que necesita información procedente de los sensores internos del vehículo. La velocidad se obtiene a partir de sensores del vehículo. Esto implica que el sistema de odometría visual no es independiente ni robusto ante fallos de los sensores internos del vehículo. El ejemplo más sencillo de esto sería en el caso de un derrape de una rueda. Los sensores internos proporcionarían un dato de velocidad erróneo que sería utilizado en el cálculo del movimiento y, por tanto, el error sería propagado por el sistema de odometría visual. Por un lado, se tiene un sistema computacionalmente viable en entornos de navegación en tiempo real, debido a las restricciones basándose en la cinemática del vehículo, pero por otra parte, no se tiene un sistema puro de odometría visual, pues depende de los sensores propioceptivos del robot.

Existen otros trabajos como los trabajos de González et al. [30, 31] que presentan una estrategia de odometría visual completamente distinta. A partir de la información obtenida por una cámara que enfo-

ca directamente hacia el suelo se calcula el movimiento de avance del vehículo, mientras que a partir de las imágenes adquiridas por una cámara enfocando directamente el horizonte se calcula el giro realizado, como si se tratase de una brújula basada en visión. Este sistema que permite su uso en condiciones de tiempo real conlleva sin embargo una restricción difícilmente compatible con una navegación en un entorno real. Esta restricción consiste en que la brújula visual, que está enfocada hacia el horizonte, no permite movilidad alguna de los objetos presentes en las imágenes capturadas, por lo que sólo sería aplicable en entornos estáticos.

El hecho de que no haya movimiento en la imagen, que puede ser inicialmente una restricción asumible en odometría visual, no es aceptable cuando el campo de visión es el horizonte. Dado que efectivamente existe la necesidad de que no existan objetos móviles en la escena visible, lo que se ha de intentar es reducir el campo de visión o campo de trabajo. Además, asumiendo que esa restricción no es creíble en un entorno real, donde existe movimiento de objetos, el campo de visión ha de ser estudiado a conciencia para que el efecto de un posible objeto móvil sea mínimo.

## 2.5. Consideraciones finales

Un procedimiento de odometría visual se fundamenta en la correcta elección de los métodos de detección y seguimiento de características, así como de una propuesta de procedimiento del cálculo del movimiento.

En cuanto a la primera parte, se han estudiado y aquí presentado los diferentes métodos existentes en la literatura. En este trabajo de investigación nos centraremos en maximizar la eficiencia de los métodos de detección de características para así minimizar el coste computacional dentro del procedimiento completo de odometría visual.

Por otra parte, la dependencia de los diferentes métodos, descriptores y *trackers* respecto del entorno de trabajo hará preciso un esfuerzo experimental que nos lleve a evaluar los diferentes métodos existentes en base a dichas condiciones y también en base a las restricciones temporales. Se tendrá que comprobar los comportamientos de detectores complejos como FAST [85], SIFT [56], SURF [3] y realizar una comparativa respecto de detectores más sencillos y con menor carga computacional como Ha-

rris [34] o Shi-Tomasi [94].

Una vez minimizado el coste computacional y maximizada la eficiencia de la detección y el seguimiento de características a partir de una secuencia de imágenes, se ha de realizar la propuesta del cálculo del movimiento realizado. En esta sección, se han explicado detalladamente las diferentes aproximaciones que ha habido a lo largo de los años. De estas investigaciones se han obtenido conceptos que se consideran útiles para esta propuesta y se han descartado soluciones que se han considerado no-óptimas.

En este trabajo de investigación, se tomará la idea de la estimación del movimiento a partir del cálculo del Centro Instantáneo de Rotación, presentado por Scaramuzza [91]. Sin embargo, nos hemos de enfrentar a que el enfoque presentado no era una propuesta exteroceptiva pura, puesto que dependía de la información proporcionada por sensores internos. El objetivo que se pretende es lograr un procedimiento de odometría visual que, aunque se pueda fusionar con información de otros sensores, se muestre independiente de sensores internos en caso de necesidad; es decir, que el trabajo propuesto se pueda considerar una propuesta exteroceptiva pura.

A la vez, se pretende logra un procedimiento de cálculo de movimiento reduciendo su coste computacional. Para ello, se descartará el uso de RANSAC -utilizado comúnmente en la literatura- y en cambio, se aplicará el uso de cálculos simples y filtrados estadísticos. Milella y Siegwart [64] realizaron un filtrado novedoso y una eliminación de valores anómalos diferente. Utilizaron un filtro basado en la mediana y la desviación típica a la hora de la detección y seguimiento de las características. Este tipo de técnicas de filtrado, más básicas y, por tanto, más rápidas en tiempos de ejecución que la utilización de RANSAC tanto para el cálculo del movimiento como para la eliminación de valores anómalos, resultó ser válido en dicho trabajo. Al igual que el uso de un filtro estadístico sencillo resulta útil y rápido en partes del proceso de odometría visual, resulta pues interesante estudiar qué ocurre si es utilizado en otras partes del proceso, principalmente, en el propio cálculo del movimiento. Esto permitiría reducir el coste computacional y temporal sin tener consecuencias en la calidad de los resultados, lo que redundaría en un aumento de la eficiencia del procedimiento propuesto.

## Capítulo 3

# Procedimiento de Odometría Visual propuesto

En este capítulo se explica con detalle cada uno de los pasos que tienen lugar en el procedimiento de odometría visual propuesto, centrándonos en los aspectos innovadores de nuestra propuesta y aportando las ventajas respecto a los trabajos explicados con anterioridad en el estado del arte.

### 3.1. Introducción

Uno de los objetivos de esta Tesis es la propuesta de un procedimiento de Odometría Visual que mejore los problemas ya comentados en procedimientos similares detallados en el estado del arte. En base a las ventajas e inconvenientes de los trabajos realizados durante los últimos años, se pretende proponer un procedimiento de Odometría Visual puramente exteroceptivo con capacidades de tiempo real.

Se necesitará extraer la información de la secuencia de imágenes que se toma como entrada y por tanto localizar los puntos característicos de cada una de las imágenes capturadas y realizar el seguimiento de los mismos a lo largo de las imágenes pertenecientes a la secuencia.

En la literatura, existen multitud de métodos de detección de características en una imagen [85, 56, 3, 34, 94]. Será preciso realizar un



esfuerzo experimental para encontrar el método más adecuado y eficiente en base también a las restricciones temporales que se plantean en este trabajo.

Una vez extraída la información a partir de las imágenes que se toman como datos de partida, se procederá a calcular el estado del robot y, por tanto, la estimación del movimiento que ha tenido lugar. Tomaremos la idea del Centro Instantáneo de Rotación del trabajo de Scaramuzza [91]. Sin embargo, en este caso no utilizaremos otros sensores internos, sino que simplemente se tomará la información obtenida del sistema de visión, por lo que se considera una propuesta puramente exteroceptiva. Éste es uno de los pilares de la propuesta investigadora y será detallado en la siguiente sección.

Se partirá de un modelo teórico donde asumimos como hipótesis lo que hemos denominado orientación robocéntrica estática del suelo, es decir, se puede asumir que el suelo tiene una posición estática respecto a la cámara y por tanto se puede identificar de forma unívoca los puntos de la cámara respecto del suelo. Como se mostrará, esta hipótesis nos permitirá evitar el uso de sensores internos. De esta forma, la detección de las características de la imagen que se utilizarán estarán condicionadas a la adecuación al modelo de dichas características.

El modelo teórico del que se parte permitirá calcular las velocidades sin necesidad de usar sensores internos.

Durante el proceso de estimación del movimiento se realizará filtrados de datos eliminando valores anómalos que mejoren el cálculo del estado definitivo. Se aplicará un clasificador borroso que defina el estado posible del robot para así realizar el cálculo del movimiento en base dicho estado: parada, giro, avance, etc. Esto nos permitirá evitar errores debido a las peculiaridades de cada uno de los estados.

Esta propuesta parte también de una restricción temporal que permitirá implementarse en sistemas de tiempo real. Para ello, se ha de buscar y optimizar la eficiencia de todos los elementos de proceso que formarán parte del procedimiento. Esta restricción implica en muchas ocasiones la realización de una experimentación que resalte los beneficios de métodos con poca carga temporal buscando así optimizar la eficiencia, no sólo en base a los resultados sino también en base a su coste computacional y temporal.

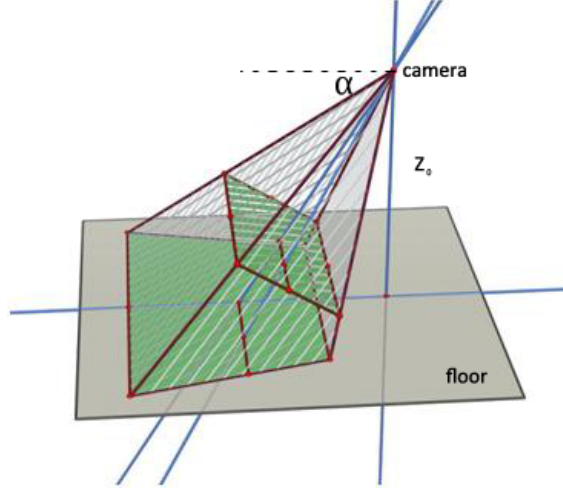


Figura 3.1: Situación de la cámara

### 3.2. Modelo teórico para el desarrollo

Se considerará la proyección en el plano del objetivo de la cámara  $\bar{x} \in \mathbb{R}^2$  de los puntos del plano del suelo del robot  $x \in \mathbb{R}^2$  (punto del plano del espacio de trabajo), como se observa en la Figura 3.1:

$$x \rightarrow \bar{x} = P(x)$$

En este sentido, se introduce una hipótesis de trabajo, por la cual se considera que el plano del suelo del robot móvil ocupa una posición fija respecto de la cámara, como se observa en la Figura 3.2 (orientación robocéntrica estática del suelo). Es evidente que se puede definir la proyección inversa.

$$\bar{x} \rightarrow x = P^{-1}(\bar{x})$$

Se considera que dicha proyección dependerá de la colocación  $(x, y, z)$  respecto del sistema de referencia fijo del robot así como de los ángulos de orientación (o *Pitch*, *Yaw*, *Roll*)  $(\alpha, \psi, \phi)$ , esto es,  $P \approx f((x, y, z, \alpha, \psi, \phi))$ . Para facilitar los cálculos, se asume que la cámara está situada sobre el centro de giro del robot móvil, a una altura  $H$  con una inclinación de

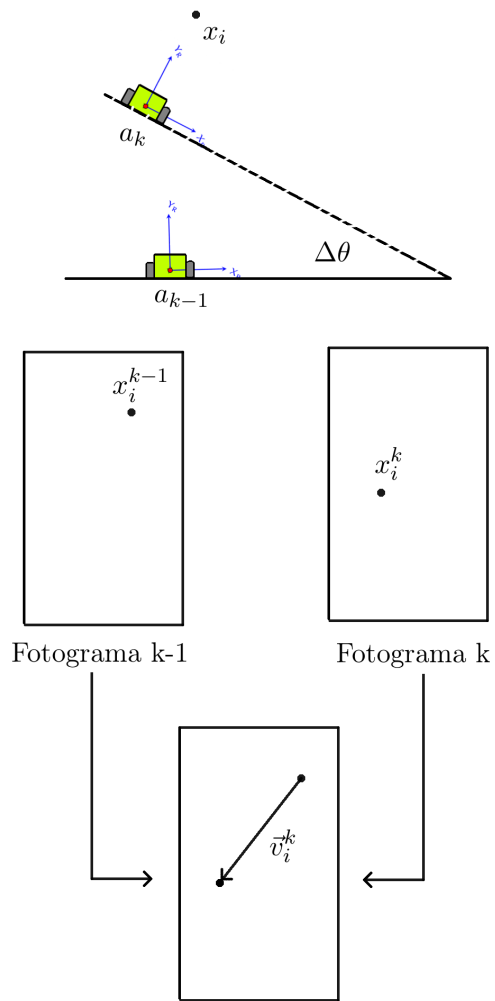


Figura 3.2: Orientación robocéntrica estática del suelo

$-\pi/4$ , por lo que dichos valores son constantes:

$$x = 0, y = 0, z = Z_0, \alpha = -\pi/4, \psi = 0, \phi = 0$$

En el apartado 3.4.1 se dan las expresiones analíticas de esta proyección. Ésta constituye un elemento clave para reducir la dimensión del problema o, en definitiva, posibilita plantear una solución monocular al procedimiento de odometría visual. Asimismo, se puede considerar que variando el valor de  $\alpha$  el área de espacio de trabajo que se puede proyectar en el plano de la cámara puede ser variado.

En situaciones reales, cabe considerar la existencia de puntos fuera del plano del suelo, asociados a los baches o irregularidades del terreno, o el hecho de que dicho plano esté orientado de otra forma (diferente a  $(\alpha, \psi, \phi) = (\pi/4, 0, 0)$ ) por la inclinación del vehículo móvil. Nuestra hipótesis de trabajo es que estas circunstancias se puedan considerar como fuentes de ruido en el proceso de medida. La amplitud y efecto de este ruido se podrá verificar en la parte experimental de este trabajo de investigación.

Sean  $\{\bar{x}_1^{k-1}, \dots, \bar{x}_j^{k-1}\}$  el conjunto de características detectables en el frame capturado  $(k-1)$ , array  $\mathcal{C}_1$ , que son proyecciones en el plano de la cámara de  $\{x_1^{k-1}, \dots, x_j^{k-1}\}$  pertenecientes al espacio de trabajo. De forma análoga, sean  $\{\bar{x}_1^k, \dots, \bar{x}_j^k\}$  las características correspondientes en el frame posterior  $k$ , array  $\mathcal{C}_2$ , de los puntos  $\{x_1^{k-1}, \dots, x_j^{k-1}\}$ . Se asume que el procedimiento de emparejamiento funciona de forma correcta, esto es,

$$\forall i \in \{1, \dots, j\}, \bar{x}_i^{k-1} \text{ emparejado con } \bar{x}_i^k$$

Si ahora consideramos las proyecciones inversas de dichos puntos, esto es:

$$x_i^k = P^{-1}(\bar{x}_i^k)$$

$$x_i^{k-1} = P^{-1}(\bar{x}_i^{k-1})$$

Es evidente que si el procedimiento de correspondencia es correcto, tanto  $x_i^k$  como  $x_i^{k-1}$  harán referencia al mismo punto del espacio (plano del suelo)  $\mathbf{x}_i$ . Ello se puede observar esquemáticamente en la Figura 3.2.

Así, como se puede apreciar en dicha figura, el vector de velocidad de movimiento del objeto  $i$  del suelo se corresponde, con la hipótesis evidente de que el objeto está fijo, con el movimiento del robot:

$$\vec{v}_i^k = \overrightarrow{[x_i^{k-1}, x_i^k]} \text{ con } \delta t \text{ suficientemente pequeño.}$$

Por tanto, en el instante  $k$ , para cada característica del suelo  $i$ , se puede considerar que podemos obtener una medida (con ruido)  $\vec{v}_i^k$  de la velocidad del robot.

El siguiente paso es considerar cómo se obtiene la estimación de esta variable pues como se ha comentado, aunque el ruido es importante, es esperable que se disponga de un número elevado de medidas. Este hecho se ha comentado en el apartado correspondiente a los procedimientos de flujo óptico.

En vez de utilizar el vector de velocidad  $\vec{v}$ , se considerará (como es habitual en robótica móvil) la pareja de valores  $(CIR, \omega)$ , esto es, los valores instantáneos del centro de giro,  $CIR$ , y velocidad angular,  $\omega$ . En la Figura 3.3 se observa, que en el instante  $k$ , cada instancia de  $\vec{v}_i^k$  se corresponde con una medida  $(\hat{CIR}^k, \hat{\omega}^k)$  de ambas variables. En la figura se puede observar cómo se puede realizar la estimación del valor del centro de giro,  $O$  en cada instante de tiempo. En nuestro caso se considera el filtro  $1-\sigma$ :

En las próximas secciones se realizará una descripción de nuestra propuesta de esta estimación. En ella cabe señalar que el coste computacional de estas estimaciones se puede considerar despreciable comparada con los costes asociados a la evaluación del flujo óptico. Asimismo, se puede considerar que se trata de una medida completamente exteroceptiva lo que hace que pueda ser perfectamente compatible, complementaria e independiente de los planteamientos odométricos clásicos en la robótica móvil. Las velocidades se calculan sin necesidad de sensores internos del robot.

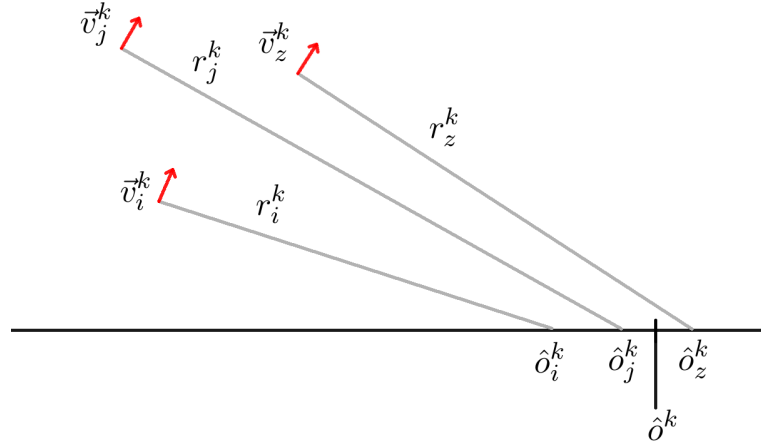


Figura 3.3: Cálculo del Centro Instantáneo de Rotación

### 3.3. Detección y seguimiento de características

#### 3.3.1. Detección de puntos característicos en una imagen

Se buscará extraer información de las imágenes capturadas por el sistema de visión. En definitiva, lo que se busca es obtener los puntos característicos de las diferentes imágenes, que serán la información de entrada al proceso de estimación del movimiento y por tanto al proceso del cálculo del estado del robot.

El primer paso es obtener un par de imágenes capturadas en dos instantes de tiempo consecutivos,  $(I_1, I_2)$ . Las imágenes son capturadas por una cámara que se encuentra sobre el robot apuntando hacia el suelo con un ángulo  $\alpha$  y una altura  $Z_0$ , como se muestra en la figura 3.1.

De esta manera, el campo de visión queda limitado y se logra una mayor consistencia espacial puesto que hay mas similitudes en los objetos que aparecen en la imagen cuando el campo de visión es más pequeño. Del mismo modo, se obtiene una iluminación más uniforme en la imagen. Si la altura a la que se encuentra la cámara,  $Z_0$ , es suficientemente baja, el campo de visión será reducido y por tanto, el riesgo de cambios de iluminación dentro de la imagen se reduce. Sin embargo, seguirán existiendo problemas con brillos, que son muy típicos en suelos de interiores

debido a la iluminación artificial, así como problemas con sombras de objetos en el exterior.

Los parámetros de la cámara, tanto intrínsecos como extrínsecos, forman parte también de las entradas del procedimiento. Los valores extrínsecos serán la altura a la que se encuentra situada la cámara,  $Z_0$ , y el ángulo de inclinación de la misma con respecto al suelo,  $\alpha$ . Los valores intrínsecos serán la distancia focal y el tamaño del sensor CCD que serán también utilizados durante el proceso de cálculo.

Para la detección y extracción de características de las imágenes se ha utilizado el detector de características de Shi-Tomasi [94]. Al ser un problema de ejecución en tiempo real, es importante que el procedimiento de odometría visual sea rápido y minimice el gasto temporal de cada paso. Los detectores de esquinas o puntos tienen un coste computacional inferior al de otros detectores de características. Además, en nuestro caso, la cámara se mantiene fija a lo largo del tiempo, por lo que no existe un problema de escalado en las imágenes. Del mismo modo, no se tendrán cambios abruptos entre dos frames consecutivos de la secuencia de imágenes capturada siempre que el tiempo de ejecución de cada iteración del algoritmo sea pequeña. Por lo tanto, basándonos en las conclusiones anteriores, utilizar un detector de puntos o esquinas con alto grado de repetibilidad, como el propuesto por Shi-Tomasi, resulta ser extremadamente beneficioso en el correcto planteamiento del problema.

Los puntos característicos sobre los que se realizará el seguimiento en los siguientes frames se tomarán dispersados por todo el espacio de visión de la cámara. De esta manera, se evitan concentraciones de características en puntos concretos de la imagen, que podrían llevar a una mala interpretación del movimiento, así como errores ocasionados por la falta de variabilidad de texturas. Por tanto, los errores que ocurren debido a los brillos y sombras o, simplemente, errores propios de la detección de características deberían ser filtrados en los siguientes pasos del procedimiento y los valores correctos deberían prevalecer.

La parte de abajo de la imagen no se toma en cuenta. Esto se hace para evitar la detección de puntos característicos que en un futuro puedan llevarnos a error. Según la posición de la cámara (apuntando hacia el suelo), los puntos inferiores de la imagen son los más cercanos al robot y por tanto, aquéllos cuyo movimiento será mayor. Además, estos puntos serán los primeros en desaparecer del campo de visión en

futuros instantes de tiempo, por lo que estos puntos, aún detectándose correctamente, no servirían para el seguimiento en futuras capturas de la secuencia de imágenes. Eliminando del espacio de muestreo la región inferior de la imagen se logra un doble beneficio: por un lado se evitan posibles fuentes de error y por otro se logra una reducción del tiempo de ejecución. Esto facilita la viabilidad de nuestro procedimiento para una ejecución en tiempo real.

---

**Algorithm 1** Detección y extracción de características
 

---

```

1: procedure DETECCIONCARACTERÍSTICAS( $I_1$ )
2:    $\mathcal{S} \leftarrow \text{secciones}(I_1)$ 
3:    $\mathcal{C}_1 \leftarrow \text{shiTomas}(I_1, \mathcal{S}, \text{numeroDePuntos})$ 
4:   return  $\mathcal{C}_1$ 
5: end procedure
  
```

---

En el algoritmo 1 se muestra el proceso de detección y extracción de características. Se calculan primero las secciones en las que se divide la imagen para que el muestreo de características sea uniforme por todo el espacio de visión.  $\mathcal{S}$  es un *array* que identifica el número de regiones de la imagen y las coordenadas de cada una de ellas. En nuestro caso, el muestreo se realiza eliminando una región en la parte inferior de la imagen y el resto se divide en 4 regiones iguales. El número de puntos característicos, *numeroDePuntos*, se distribuye uniformemente por cada una de las regiones en las que se divide la imagen.  $\mathcal{C}_1$  es un *array* que contiene las coordenadas de cada punto característico que ha sido detectado.

### 3.3.2. Seguimiento de las características

La selección de características y el seguimiento de dichos puntos característicos en la secuencia de imágenes es un paso importante dentro del procedimiento de cálculo del flujo óptico entre dos imágenes. Hay dos enfoques fundamentales. El primer enfoque consiste en encontrar los puntos característicos en la primera imagen y realizar un seguimiento de dichos puntos a lo largo de las siguientes imágenes. El segundo enfoque se basa en encontrar puntos característicos en cada una de las imágenes para posteriormente localizar posibles similitudes entre los descriptores de los puntos de diferentes imágenes.



En este trabajo, se utiliza la implementación piramidal del algoritmo Lukas-Kanade propuesto por Bouguet [7]. El proceso de detección de características se realiza en un principio y, una vez encontrados los puntos característicos de la primera imagen, se realiza un seguimiento de los mismos en las siguientes imágenes. El uso de este algoritmo, implementado en numerosas bibliotecas de visión artificial, es más eficiente computacionalmente que el algoritmo Lukas-Kanade original [58]. Sin embargo, no proporciona la propiedad de consistencia que sí está presente en la implementación del detector de características de Lukas-Kanade-Tomasi [106]. En resumen, consiste en comparar los valores de intensidad de una ventana que rodea al punto característico en la primera imagen en la que éste aparece con los valores de intensidad de una ventana alrededor del punto característico de la imagen actual. Esto se hace para corregir posibles cambios que tienen lugar en la imagen debido al movimiento realizado. De esta manera, los puntos característicos de una imagen pueden ser detectados en imágenes separadas en el tiempo aunque se haya producido un desplazamiento grande. En nuestro trabajo, el detector de características debe ser ejecutado tras cortos períodos de tiempo, y por tanto desplazamientos pequeños, puesto que nuestro método no tiene ninguna corrección ante desplazamientos grandes. En cambio, el algoritmo propuesto por Bouguet [7], y la implementación que se utiliza en este trabajo, que está presente en la biblioteca OpenCV de Intel, maneja de manera más correcta la detección de los puntos característicos cerca de los bordes de la imagen y es más eficiente en tiempos de ejecución. Esta última característica es extremadamente importante en nuestro deseo de lograr una procedimiento que sea viable para ser ejecutado en tiempo real.

---

**Algorithm 2** Seguimiento de las características

---

```

1: procedure SEGUIMIENTOCHARACTERISTICAS( $I_1, I_2, C_1$ )
2:    $C_2 \leftarrow pyr\_Lukas\_Kanade(I_2, C_1)$ 
3:   return  $C_2$ 
4: end procedure

```

---

El Algoritmo 2 muestra el procedimiento del seguimiento de puntos característicos. La entrada del algoritmo son las dos imágenes sobre las que se realiza el cálculo de odometría visual,  $I_1$  and  $I_2$  junto con los puntos característicos de la primera imagen,  $C_1$ , que se han obtenido previamente. Posteriormente, los puntos obtenidos de la primera imagen se localizan en la segunda imagen quedando almacenados en  $C_2$ , que es un

array con las coordenadas de dichos puntos característicos en la segunda imagen.

### 3.4. Cálculo del estado

El segundo bloque del procedimiento propuesto se centrará en la estimación del movimiento y, por tanto, en el cálculo del estado del robot en base a ese movimiento estimado. Se toman como entradas los puntos característicos de la secuencia de imágenes capturadas por el sistemas de visión. Un primer paso será realizar la transformación de los puntos situados en la imagen, en 2D, a puntos de la realidad, 3D. En ese momento, se podrá aplicar el modelo teórico presentado en la sección anterior, 3.2.

#### 3.4.1. Proyección de un punto de la imagen

Una vez que se han obtenido los puntos característicos de la primera imagen y se han localizado en la segunda imagen, es necesario transformar los valores referenciados sobre la imagen en dos dimensiones a coordenadas en tres dimensiones referenciadas en el sistema de referencia global. Utilizando técnicas de geometría descriptiva es posible proyectar un punto de la imagen  $(x, y)$  en el mundo real, dando un punto  $(X, Y, Z)$ .

El campo de visión es limitado debido a que la cámara se encuentra enfocada hacia el suelo con un ángulo  $\alpha$  y elevada  $Z_0$  milímetros. El campo de visión de la cámara está formado íntegramente por puntos correspondientes a la superficie del suelo, lo que significa que todos los puntos que se observan en una imagen tienen una dimensión común. La altura de todos los puntos de la imagen se considera  $Z = 0$ .

---

**Algorithm 3** Paso de coordenadas 2D a coordenadas 3D

---

```

1: procedure 2DA3D( $\mathcal{C}_1, \mathcal{C}_2, \alpha, Z_0$ )
2:    $(\lambda_X, \lambda_Y, CCD_X, CCD_Y) \leftarrow get\_camera\_values()$ 
3:    $\mathcal{R}_1 \leftarrow calcularCoordinadas3D(\lambda_X, \lambda_Y, CCD_X, CCD_Y, \alpha, Z_0, \mathcal{C}_1)$ 
4:    $\mathcal{R}_2 \leftarrow calcularCoordinadas3D(\lambda_X, \lambda_Y, CCD_X, CCD_Y, \alpha, Z_0, \mathcal{C}_2)$ 
5:   return  $(\mathcal{R}_1, \mathcal{R}_2)$ 
6: end procedure

```

---

El algoritmo 3 muestra los cálculos necesarios para realizar la proyección de un punto de la imagen y, por tanto, obtener las coordenadas de dicho punto de la imagen en el mundo real. Es necesario conocer determinados parámetros de la cámara: distancia focal, tamaño del sensor CCD, posición de la cámara y el ángulo de inclinación hacia el suelo.  $\mathcal{C}_1$  y  $\mathcal{C}_2$  son los arrays que contienen las coordenadas de los puntos característicos del par de imágenes que se está evaluando.  $\mathcal{R}_1$  y  $\mathcal{R}_2$  contienen la proyección en el mundo real de los puntos característicos de cada imagen. La transformación se realiza siguiendo las ecuaciones 3.1 y 3.2,

$$Y = \frac{(\lambda_Y \cdot Z_0 \cdot \sin \alpha) + (y_i \cdot Z_0 \cdot \cos \alpha) + (y_i \cdot \lambda_Y)}{(\lambda_Y \cdot \cos \alpha) - (y_i \cdot \sin \alpha)} \quad (3.1)$$

$$X = \frac{(x_i \cdot Y \cdot \sin \alpha) + (x_i \cdot Z_0 \cdot \cos \alpha) + (x_i \cdot \lambda_X)}{\lambda_X} \quad (3.2)$$

donde  $(x_i, y_i)$  son las coordenadas de un punto en la imagen, en milímetros (en el sensor CCD de la cámara);  $(X, Y, Z)$  son las coordenadas de dicho punto en el sistema de referencia global, en milímetros;  $(\lambda_x, \lambda_y)$  son las componentes  $x$  e  $y$  de la distancia focal de la cámara, en milímetros, obtenida tras realizar un proceso de calibrado oportuno.;  $\alpha$  es el ángulo de inclinación hacia el suelo en el que se encuentra la cámara; y  $Z_0$  es la altura respecto del suelo a la que se encuentra la cámara. De esta manera, el campo de visión de la cámara es limitado y medible. La coordenada  $Z$  de cada punto característico detectado se considerará siempre  $Z = 0$ , puesto que se asume que todos los puntos pertenecientes al campo de visión se encuentran sobre la superficie del suelo.

### 3.4.2. Eliminar valores anómalos

Una vez que se tienen los puntos característicos expresados en base al sistema de referencia global y con sus tres dimensiones, se procede a realizar los cálculos necesarios para obtener el movimiento real que ha tenido lugar entre las dos imágenes que se tomaron como entrada del procedimiento completo. El primer procesamiento previo que se realiza es la eliminación de los posibles errores en los datos de entrada. Esta información errónea puede ser causada por fallos en el proceso de detección de características o por la aparición de objetos extraños en la imagen. El primer tipo de error clasificado puede ser la aparición de sombras y

brillos en la imagen que simulen un cierto movimiento aparente. Otro tipo de error es la presencia de un objeto sobre la superficie del suelo, que no admitiría la condición comentado en previos apartados de que la altura de dicho objeto sea  $Z = 0$ . Esto nos llevará a la aparición de puntos característicos de la imagen que no muestren de forma correcta el movimiento real que ha tenido lugar. El algoritmo 4 muestra el proceso de filtrado que se realiza.

---

**Algorithm 4** Eliminar valores anómalos
 

---

```

1: procedure ELIMINARANOMALOS( $\mathcal{R}_1, \mathcal{R}_2$ )
2:    $\mathcal{D} \leftarrow \mathcal{R}_2 - \mathcal{R}_1$ 
3:    $(\mathcal{F}_1, \mathcal{F}_2) \leftarrow \text{anomalos\_modulo}(\mathcal{D})$ 
4:    $\mathcal{D} \leftarrow \mathcal{F}_2 - \mathcal{F}_1$ 
5:    $(\mathcal{F}_1, \mathcal{F}_2) \leftarrow \text{anomalos\_orientacion}(\mathcal{D})$ 
6:   return  $(\mathcal{F}_1, \mathcal{F}_2)$ 
7: end procedure

```

---

Se calcula primeramente el vector de desplazamiento para cada punto característico de la primera imagen que coincide con un punto de la segunda imagen,  $\mathcal{D}$ . Se calcula el módulo y la orientación de dicho vector por cada punto característico. Los vectores que tengan un valor de módulo o un valor de orientación anómalo se eliminan. En nuestro caso, se utiliza un filtro  $1\sigma$  para detectar los valores anómalos. De esta manera, los puntos característicos que se tendrán en cuenta en futuros cálculos serán únicamente aquéllos que satisfagan que el módulo y la orientación de sus vectores de desplazamiento no sean valores anómalos. Los puntos característicos detectados en la primera y segunda imagen correctos se almacenan en los arrays  $\mathcal{F}_1$  and  $\mathcal{F}_2$ , respectivamente.

### 3.4.3. Cálculo del Centro Instantáneo de Rotación

Tras la eliminación y filtrado de los valores anómalos, en módulo y orientación, del conjunto de vectores de desplazamiento obtenidos por cada uno de los puntos característicos de la imagen, los valores restantes se utilizan para calcular el movimiento que se ha producido en el tiempo transcurrido entre la captura de ambas imágenes. Para ello, se considera que cualquier movimiento que el robot puede realizar puede definirse como un arco de ángulo  $\theta$  y radio de giro  $R$ . De esta manera, un movimiento en línea recta se puede considerar también como una

curva donde el radio de giro  $R = \infty$ . En el otro extremo, para un robot con configuración diferencial, como es el caso, un giro sobre sí mismo se considerará como una curva cuyo radio de giro  $R = 0$ .

Para calcular el radio de giro  $R$  es fundamental obtener el punto sobre el que el robot realiza el giro en cada instante de tiempo, esto es el Centro Instantáneo de Rotación (CIR). En el algoritmo 5, se explica el proceso implementado para el correcto cálculo del Centro Instantáneo de Rotación. El conjunto de características de la primera imagen,  $\mathcal{F}_1$ , y los puntos correspondientes en la segunda imagen,  $\mathcal{F}_2$ , sirven de entrada del algoritmo.

---

**Algorithm 5** Calcular CIR
 

---

```

1: procedure CALCULARCIR( $\mathcal{F}_1, \mathcal{F}_2$ )
2:    $CIRCandidatos \leftarrow \text{calcular\_CIR\_Candidatos}(\mathcal{F}_1, \mathcal{F}_2)$ 
3:    $CIR_F \leftarrow \text{filtro\_sigma}(CIRCandidatos)$ 
4:    $CIR \leftarrow \text{moda}(CIR_F)$ 
5:    $\rho \leftarrow \text{calcular\_modulo}(\mathcal{F}_1, \mathcal{F}_2, CIR_F)$ 
6:   return ( $CIR, \rho$ )
7: end procedure

```

---

A partir de cada punto característico se calcula un vector de desplazamiento, que indica el movimiento que dicho punto ha sufrido y cada vector de desplazamiento proporciona un punto de rotación (ICR) posible. De esta manera, se obtendrá un conjunto de puntos ICR candidatos. La Figura 3.4a) muestra un conjunto de vectores de desplazamiento. El trapecio pintado en azul identifica el campo de visión de la cámara.

Cada uno de esos vectores queda identificado por los puntos  $(X_1, Y_1, Z_1)$ , que es la posición del punto característico en la primera imagen, y  $(X_2, Y_2, Z_2)$ , que es la localización del punto característico en la segunda imagen. Para ambos puntos  $Z_1 = Z_2 = 0$ , ya que se considera que todos los puntos pertenecientes al campo de visión de la cámara están situados en la superficie del suelo. Se calcula la línea perpendicular a cada vector de desplazamiento que pasa por el punto medio del vector para obtener el CIR propuesto por cada uno de los vectores de desplazamiento que se tienen. El punto donde esta perpendicular interseca el eje de rotación del robot será el CIR definido por dicho vector de desplazamiento. En la Figura 3.4b) se muestra un ejemplo de los diferentes CIR candidatos que han sido obtenidos a partir de cada uno de los vectores de desplazamiento.

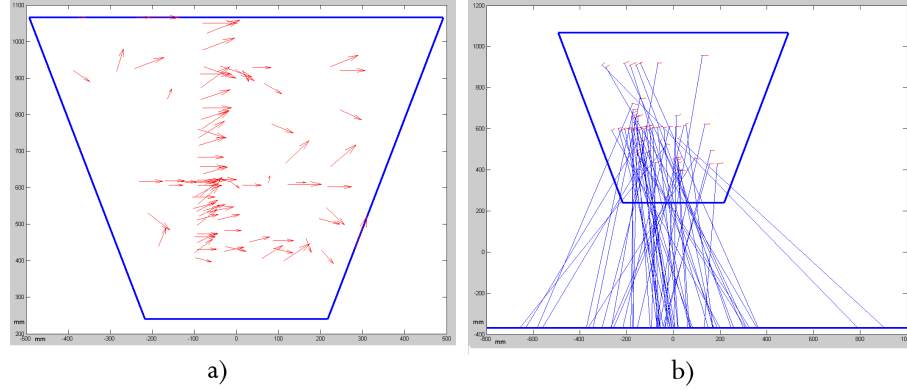


Figura 3.4: a) Vectores de desplazamiento para un par de imágenes, b) Puntos CIR candidatos

El eje de rotación del robot está definido por la configuración del robot utilizado. En el caso de un robot con configuración diferencial, como es nuestro caso, el eje de rotación será la línea que forman las dos ruedas motrices del robot.

Tras obtener un conjunto de puntos CIR candidatos, se dispondrá de una distribución de puntos a lo largo de la línea formada por el eje de rotación. El valor que devuelve el algoritmo debe ser un único punto CIR que definirá la curva que ha tenido lugar. Primeramente, se realiza un filtrado  $1\sigma$  que elimina valores de puntos anómalos. Esto se puede considerar como un segundo proceso de filtrado de valores anómalos, ya que la eliminación de los vectores de desplazamiento cuyo valor en módulo u orientación era anómala ya ha sido realizada en el paso anterior. Este segundo proceso de filtrado reduce aún más los posibles errores o valores ilógicos que se hallen en el conjunto de vectores de desplazamiento. Una vez que se tiene una distribución libre de errores de los puntos CIR candidatos, se proporciona un valor CIR final como resultado del algoritmo.

Se construye un histograma de puntos CIR como se muestra en la Figura 3.5 para calcular el valor final del CIR. Este valor final será el punto  $(X_{ICR}, Y_{ICR})$ , de forma que  $X_{ICR}$  será la moda agrupada calculada siguiendo la ecuación 3.3:

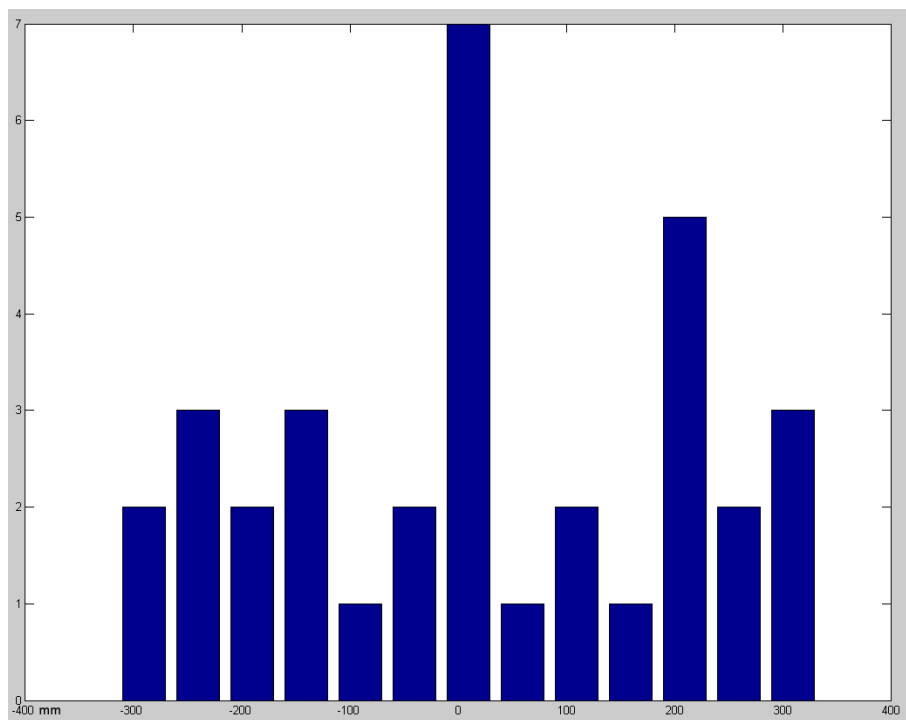


Figura 3.5: Histograma de CIR candidatos

$$M = L_{i-1} + \left( \frac{D_1}{D_1 + D_2} \right)_i \quad (3.3)$$

donde  $L_{i-1}$  es el nivel anterior a la clase modal,  $D_1$  es la diferencia entre la frecuencia de la clase modal y la frecuencia de la clase pre-modal, y  $D_2$  es la diferencia entre la frecuencia de la clase modal y la frecuencia de la clase post-modal. El valor  $Y_{ICR}$  es el valor  $Y$  del eje de rotación del robot según el sistema de referencia global.

El valor del módulo del vector de desplazamiento final,  $\rho$ , será obtenido de la misma manera. Los vectores que han proporcionado valores anómalos durante el cálculo del Centro Instantáneo de Rotación son también eliminados cuando se calcula el módulo final. Al final de este proceso se habrán calculado dos valores que definen el movimiento que ha tenido lugar:  $CIR$ , que es el punto sobre el que ocurre la rotación, y  $\rho$ , que es el módulo de desplazamiento. Con estos dos valores queda definida la curva que se ha realizado.

#### 3.4.4. Filtrado de resultados

Antes de calcular el movimiento real que ha tenido lugar entre dos imágenes capturadas se realiza un nuevo proceso de filtrado, esta vez sobre los resultados que se han dado. Este proceso se describe en el algoritmo 6. Para ello, se toman los datos del radio de giro,  $CIR$ , y el módulo del vector desplazamiento,  $\rho$ , de una ventana temporal previa. Se toman los datos de los últimos  $W$  instantes de tiempo. Al ser un procedimiento que se ejecuta en tiempo real, la distancia temporal entre un par de imágenes es del orden de los milisegundos. Por tanto, no es incorrecto pensar que un robot no puede realizar movimientos abruptos en períodos de tiempo tan pequeños. De esta manera, los valores calculados en los instantes de tiempo previos afectan al cálculo del movimiento ocurrido en el instante de tiempo actual.

Se procesan los datos de rotación calculados en una ventana de los  $W$  movimientos previos junto con los valores actuales utilizando un filtro de Kalman [44]. Esto impide que el resultado obtenido a partir de las dos capturas de imágenes actuales den como resultado un cambio de posición abrupto, y por tanto erróneo. De esta manera, este resultado incorrecto no será tenido en cuenta en el cálculo del movimiento final. Por otra



**Algorithm 6** Filtrado de resultados

---

```

1: procedure FILTRADO( $ICR_{[t-W..t]}$ ,  $\rho_{[t-W..t]}$ )
2:    $CIR_\sigma \leftarrow \text{eliminar\_anomalos\_ventana}(CIR_{[t-W..t]})$ 
3:    $\rho_\sigma \leftarrow \text{eliminar\_anomalos\_ventana}(\rho_{[t-W..t]})$ 
4:    $IIR_{filtrado} \leftarrow \text{kalmán\_filter}(CIR_\sigma)$ 
5:    $\rho_{filtrado} \leftarrow \text{kalmán\_filter}(\rho_\sigma)$ 
6:   return ( $CIR_{filtrado}$ ,  $\rho_{filtrado}$ )
7: end procedure

```

---

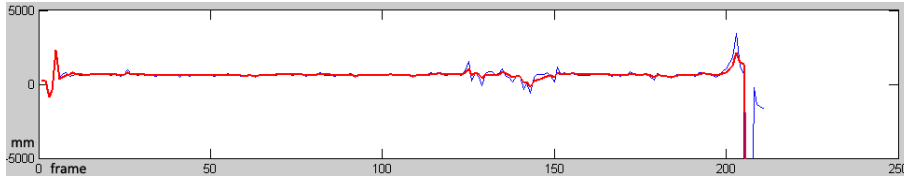


Figura 3.6: CIR filtrado (rojo) y CIR sin filtrar (azul)

parte, cuando sí se realice un cambio abrupto en la trayectoria del robot, éste será suavizado por el método de filtrado.

La Figura 3.6 muestra el  $CIR$  actual utilizado, en color rojo, y el  $CIR$  sin filtrar, en color azul. Se observa que los cambios abruptos son filtrados y, por tanto, los valores erróneos no son considerados.

### 3.4.5. Clasificador borroso

Una vez que se han calculado el centro de rotación sobre el que gira el robot,  $CIR$ , y el módulo del vector desplazamiento,  $\rho$ , se puede calcular la nueva posición del robot tras el movimiento que ha tenido lugar. Previamente, se calcula el tipo de movimiento que ha tenido lugar. Se diferencian cuatro estados: avance en línea recta, rotación sobre un punto  $CIR$ , ruido y parada.

Esta clasificación es necesaria puesto que en el caso de los avances en línea recta, el radio de giro es infinito. Esto implica que los errores se acumulan cuando se realizan movimientos en línea recta ante la imposibilidad de calcular un avance puro. En estos casos, el cálculo del  $CIR$  tiene una variabilidad pronunciada pasando de valores extremos positivos a valores extremos negativos. Esto se muestra en la Figura 3.7. En

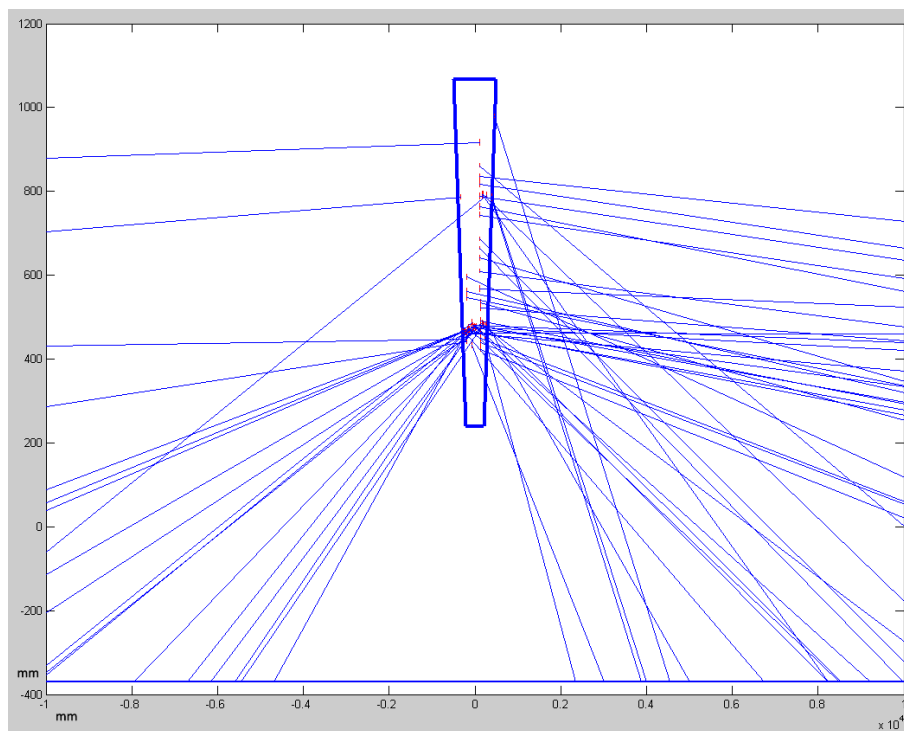


Figura 3.7: Avance en recta

los movimientos en línea recta, el avance puede tomarse como un giro a la derecha sobre un punto muy distante o como un giro a la izquierda sobre un punto muy distante, lo que implica signos opuestos. Esto lleva a errores en el cálculo del *CIR*. Es por ello por lo que los movimientos que han tenido lugar deben ser clasificados.

Se ha implementado un clasificador borroso para diferenciar el tipo de movimiento que tiene lugar. Toma dos entradas: la variabilidad del centro de rotación en los instantes de tiempo previos y el tipo de movimiento en los momentos previos. Como salida se proporciona el tipo de movimiento que se realiza en el momento actual.

Si la variabilidad del centro de rotación es grande, la probabilidad de tener un movimiento lineal es mayor. El centro de rotación en un movimiento lineal puede tener valores positivos y valores negativos muy grandes en instantes de tiempo próximos. En el caso de un giro sobre un punto concreto, la variación del centro de rotación será pequeña. Por

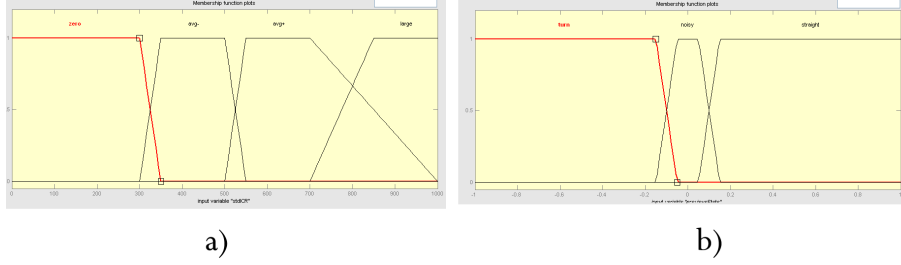


Figura 3.8: Entradas del clasificador borroso: a) Variabilidad del centro de rotación en los instantes anteriores, b) Tipo de movimiento en los instantes anteriores

tanto, una buena medida de la variabilidad del centro de rotación es la desviación estándar del *CIR* calculado durante una ventana de instantes de tiempo anteriores.

La segunda entrada del clasificador es el tipo de movimiento de los instantes de tiempo previos. De forma similar a lo comentado con el filtrado de los resultados en el apartado anterior, un cambio abrupto en el tipo de movimiento no es común. Por tanto, el tipo de movimiento que tiene lugar en el instante de tiempo que se está calculando depende también del tipo de movimientos que vienen calculándose en tiempos anteriores. Un cambio en el tipo de movimiento sólo ocurre cuando el nuevo tipo de movimiento se da de forma continuada en los instantes previos.

Las funciones de pertenencia de las entradas y sus diferentes estados se muestran en la Figura 3.8. La variabilidad del centro de rotación en los instantes de tiempo previos (Figura 3.8a)), *stdICR*, tiene cuatro posibles estados: *zero*, *avg-*, *avg+* and *large*. Las funciones de pertenencia son funciones trapezoidales que permiten especificar los límites cuando la variabilidad es pequeña (de 0 a 500mm) y que permiten introducir límites más borrosos cuando la variabilidad es mayor (más de 500mm).

El tipo de movimiento en los instantes previos (Figura 3.8b)), *previousState*, tiene tres posibles estados: *turn*, *noisy* and *straight*. Las funciones de pertenencia también son funciones trapezoidales que dan mayor peso a los valores extremos cuando el tipo de movimiento es claramente mayoritario en la ventana temporal previa. Sin embargo, cuando el tipo de movimiento no es tan obvio en los instantes previos, el valor *noisy* es

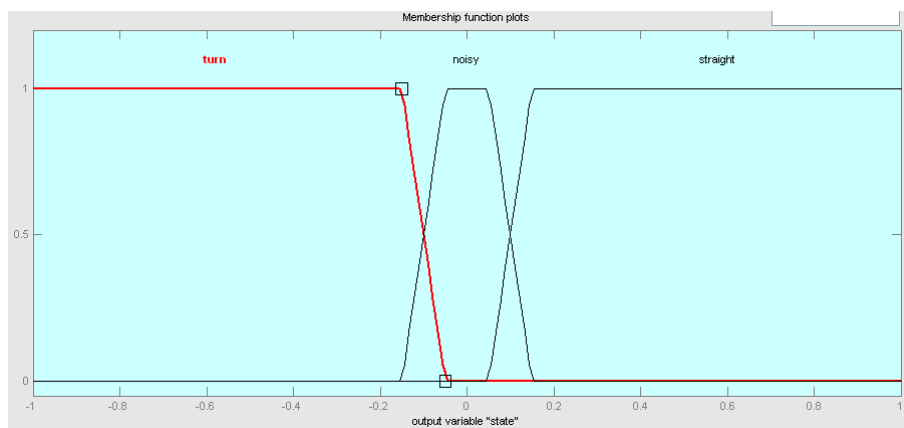


Figura 3.9: Salida del clasificador borroso

el que gana peso. El valor se calcula como la media de las salidas de la ventana temporal previa.

La Figura 3.9 muestra los estados de salida del clasificador borroso y sus funciones de pertenencia, *state*. El tipo de movimiento obtenido es *turn*, *straight* or *noisy*. Las funciones de pertenencia son funciones trapezoidales que promueven la decisión hacia uno de los dos estados semánticos posibles, excepto cuando el valor de salida es muy débil. En ese caso, la salida es *noisy*, lo que significa que el clasificador sugiere que no es posible conocer con certeza qué tipo de movimiento tiene lugar actualmente. En la próxima ejecución del procedimiento de cálculo de la odometría visual, el valor numérico de la salida será parte del array de estados previos con muy poco poder de decisión. El valor será cercano a 0 y, por tanto, no influirá mucho en el cálculo de la media de los estados previos en la siguiente iteración.

Finalmente, la Figura 3.10 muestra las reglas del clasificador borroso.

El cuarto estado posible se computa de forma previa tomando el valor del modulo del vector desplazamiento. Si es cercano a 0, se considera que el robot está en el estado de *parada* y, por tanto, ni siquiera se computa el avance y la rotación.

Una vez que se conoce el tipo de movimiento, se puede calcular ya la nueva posición del robot, puesto que se conocen el radio de giro (centro de rotación) y la distancia recorrida (el módulo del vector desplazamiento).

1. If (stdICR is zero) and (previousState is turn) then (state is turn) (1)
2. If (stdICR is zero) and (previousState is straight) then (state is noisy) (1)
3. If (stdICR is avg-) and (previousState is turn) then (state is turn) (1)
4. If (stdICR is avg-) and (previousState is straight) then (state is straight) (1)
5. If (stdICR is avg+) and (previousState is turn) then (state is turn) (1)
6. If (stdICR is avg+) and (previousState is straight) then (state is straight) (1)
7. If (stdICR is large) and (previousState is turn) then (state is noisy) (1)
8. If (stdICR is large) and (previousState is straight) then (state is straight) (1)
9. If (stdICR is zero) and (previousState is noisy) then (state is turn) (1)
10. If (stdICR is large) and (previousState is noisy) then (state is straight) (1)
11. If (stdICR is avg-) and (previousState is noisy) then (state is noisy) (1)
12. If (stdICR is avg+) and (previousState is noisy) then (state is noisy) (1)

Figura 3.10: Reglas del clasificador borroso

#### 3.4.6. Cálculo de la nueva posición

Una vez que se conoce el tipo de movimiento, el cálculo de la nueva posición del robot utilizando exclusivamente los datos obtenidos en el procedimiento de cálculo de la odometría visual se realiza siguiendo el proceso descrito en el algoritmo 7.

En caso de que el tipo de movimiento sea una rotación, el radio de giro se obtiene a partir del  $CIR$  calculado y la distancia recorrida se obtiene a partir del módulo del vector desplazamiento,  $\rho$ . El ángulo girado,  $\Delta\theta$ , se calcula sabiendo que  $\vec{\rho} = \vec{\omega} \cdot R$ . En caso de que el tipo de movimiento sea lineal, la variación de la orientación será 0, por lo que  $\Delta X = 0$  y  $\Delta\theta = 0$ . Finalmente, el cálculo de la nueva posición del robot dependerá de la posición del robot en el instante de tiempo inmediatamente anterior y del movimiento actual realizado,  $(\Delta X, \Delta Y, \Delta\theta)$ .

El sistema de referencia está centrado en el robot y la dirección de avance tiene lugar en el eje Y.

### 3.5. Consideraciones finales

Se ha propuesto un procedimiento de odometría visual puramente exteroceptivo. Se parte de un modelo teórico donde se asume como hipótesis la orientación robocéntrica del suelo, esto es, se da por supuesto que

**Algorithm 7** Calcular Nueva Posición

---

```

1: procedure CALCULARNUEVAPOSICION( $\mathcal{S}_t, \rho_{filtrado}, CIR_{filtrado}, x_{t-1}, y_{t-1}, \theta_{t-1}$ )
2:   if  $\mathcal{S}_t$  es Turn then
3:      $\Delta\theta \leftarrow \text{calcular\_angulo}(\rho_{filtrado}, CIR_{filtrado})$ 
4:      $\Delta X \leftarrow CIR_{filtrado} - CIR_{filtrado} \cdot \cos \Delta\theta$ 
5:      $\Delta Y \leftarrow CIR_{filtrado} \cdot \sin \Delta\theta$ 
6:     if leftTurn then
7:        $\Delta\theta \leftarrow -\Delta\theta$ 
8:     end if
9:   else
10:     $\Delta X \leftarrow 0$ 
11:     $\Delta Y \leftarrow \rho_{filtrado}$ 
12:     $\Delta\theta \leftarrow 0$ 
13:   end if
14:    $x_t \leftarrow (\cos \theta_{t-1} \cdot \Delta X) - (\sin \theta_{t-1} \cdot \Delta Y) + x_{t-1}$ 
15:    $y_t \leftarrow (\sin \theta_{t-1} \cdot \Delta X) + (\cos \theta_{t-1} \cdot \Delta Y) + y_{t-1}$ 
16:    $\theta_t \leftarrow \theta_{t-1} + \Delta\theta$ 
17:   return ( $x_t, y_t, \theta_t$ )
18: end procedure

```

---

el suelo tiene una posición estática respecto a la cámara y por tanto se puede identificar de forma unívoca los puntos de la cámara respecto del suelo. Esta es la hipótesis que permite evitar el uso de sensores internos para el cálculo de las velocidades.

Para la detección y extracción de características de las imágenes se ha utilizado el detector de características de Shi-Tomasi [94]. Al ser un problema de ejecución en tiempo real, es importante que el procedimiento de odometría visual sea rápido y minimice el gasto temporal de cada paso. Los detectores de esquinas o puntos tienen un coste computacional inferior al de otros detectores de características. Además, en nuestro caso, la cámara se mantiene fija a lo largo del tiempo, por lo que no existe un problema de escalado en las imágenes. Del mismo modo, no se tendrán cambios abruptos entre dos frames consecutivos de la secuencia de imágenes capturada siempre que el tiempo de ejecución de cada iteración del algoritmo sea pequeña. Por lo tanto, basándonos en las conclusiones anteriores, utilizar un detector de puntos o esquinas con alto grado de repetibilidad, como el propuesto por Shi-Tomasi, resulta ser extremadamente beneficioso en el correcto planteamiento del problema.

El movimiento se considera como un movimiento curvo entorno a un determinado punto, que será denominado Centro Instantáneo de Rotación. Para la correcta estimación del movimiento que ha tenido lugar, se calcula dicho punto de giro y el módulo de giro. Esta idea se toma del trabajo de Scaramuzza [91].

Durante el proceso de estimación del movimiento se realizan filtrados de datos eliminando valores anómalos que mejoran el cálculo del estado definitivo. Se aplica un clasificador borroso que establece cuatro posibles estados del robot: parada, giro sobre un punto, avance en línea recta -giro sobre un punto en el infinito-, ruido. En función del estado del robot en cada instante de tiempo, el cálculo del movimiento tendrá características y peculiaridades diferentes.

Se pretendía conseguir un procedimiento de odometría visual puramente exteroceptivo que cumpliera con la restricción de tiempo real. De esta manera se ha buscado optimizar la eficiencia de todos los elementos de proceso que forman parte del procedimiento. Esta restricción implica en muchas ocasiones la realización de una experimentación que resalte los beneficios de métodos con poca carga temporal buscando así optimizar la eficiencia, no sólo en base a los resultados sino también en base a su coste computacional y temporal.

## Capítulo 4

# Resultados

En este apartado se mostrarán las diversas pruebas y ejecuciones del procedimiento propuesto que se han realizado para la comprobación del correcto funcionamiento del mismo. Se detallarán los entornos donde se han realizado las pruebas, los resultados y las principales conclusiones que se obtienen de cada una de ellas.

### 4.1. Introducción

Las diferentes ejecuciones han procurado ser lo suficientemente heterogéneas para confirmar el correcto funcionamiento del procedimiento propuesto en diferentes entornos. Se han realizado pruebas en interiores y exteriores; en suelos con diferentes características, bien sea rugosos, lisos, poco texturados o con marcas. Esto último nos llevó también a realizar un estudio sobre los diferentes detectores de características existentes en la literatura en base al tipo de entorno en el que el procedimiento iba a ser utilizado. También se realizaron diversas pruebas con diferentes trayectorias y comportamientos.

De la misma manera se realizaron pruebas en el exterior utilizando diferentes vehículos, incluyendo trayectorias por carreteras utilizando vehículos a motor, que conllevan en sí un entorno de ejecución completamente diferente al de un robot, tanto por las condiciones del sistema, como por la propia característica del vehículo.





Figura 4.1: Plataforma ArViBOT

Durante las diferentes ejecuciones se tomaron las medidas de tiempo de ejecución para así comprobar la adecuación del procedimiento propuesto en un sistema de tiempo real.

La plataforma robótica utilizada durante las pruebas iniciales fue la plataforma ArViBOT del Grupo de Robótica y Sociedad de la Universidad de Salamanca, que se muestra en la Figura 4.1. Es una plataforma creada por el propio grupo de investigación equipada con dos motores EMG49 con *encoders*, dos ruedas de  $125mm$  de diámetro con una cubierta de goma de  $28mm$  de ancho y una controladora serie MD49. Por lo tanto, utiliza una configuración diferencial. La separación entre ambas ruedas es de  $365mm$ .

En cuanto al sistema de visión, consta de una cámara VIVOTEK PZ-7122 con movimiento sobre horizontal, *pan*, vertical, *tilt*, y con capacidad de *zoom*  $10\times$ . Proporciona imágenes de dimensiones  $704 \times 576$  píxeles con una tasa de 25 imágenes por segundo mediante un cliente-servidor RTSP. El procesador sobre el que se realizan las pruebas es un Intel

Core 2 Duo a 3.00GHz y con 4GB de memoria RAM.

## 4.2. Resultados del procedimiento de Odometría Visual

Se muestran en esta sección las pruebas y los resultados obtenidos utilizando únicamente el procedimiento de cálculo de odometría visual. Primeramente se muestra los resultados obtenidos sin realizar ningún tipo de fusión sensorial y basando el cálculo de la posición del robot en movimiento únicamente en la información proporcionada por el sistema de visión del robot.

### 4.2.1. Resultados en interiores

Las pruebas iniciales en interiores fueron realizadas en el edificio de la Facultad de Ciencias de la Universidad de Salamanca. Este edificio tiene entornos claramente diferenciados con mayor o menor grado de iluminación y con diferentes tipos de baldosas en el suelo dependiendo de la zona del edificio en el que se encuentre el robot. Las pruebas se llevaron a cabo en diferentes momentos del día de forma que se lograra una mayor generalización de los resultados, debido a las diferentes iluminaciones provocadas por la luz artificial o por la luz natural.

De la misma manera, a lo largo del día hay momentos con mayor afluencia de personas por los pasillos del edificio lo que implica la aparición o no de objetos extraños en el campo de visión como pueden ser pies de personas. Si bien, más allá de esas apariciones, implica también la presencia de cambios en la luminosidad y por tanto una fuente de errores debido a la aparición de sombras provocadas por el movimiento de la gente. Estos valores son valores anómalos que el sistema ha de ser capaz de eliminar o al menos mitigar lo máximo posible sus consecuencias en el resultado final.

La Figura 4.2 muestra diferentes imágenes capturadas durante los recorridos realizados en el interior del edificio. Las figuras 4.2a) y 4.2b) muestran los diferentes tipos de baldosas que podemos encontrar en función de la zona del edificio en la que se encuentre el robot. El sistema se ha probado en diferentes entornos con diferentes texturas de forma que

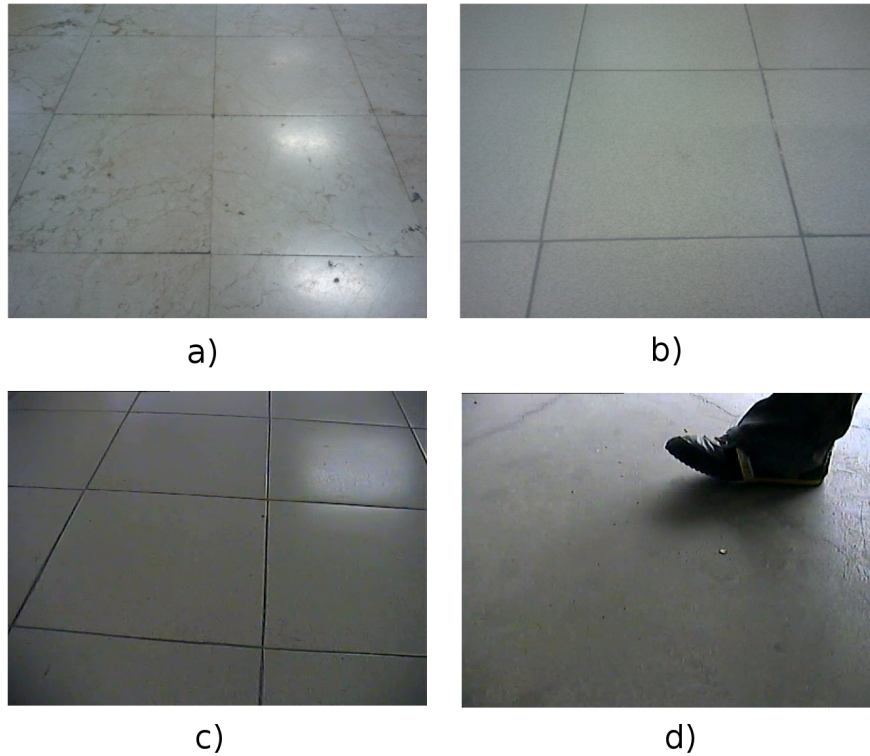


Figura 4.2: a) Baldosas del Hall de la Facultad con brillos. b) Baldosas de los pasillos. c) Baldosas de los pasillos con sombras y brillos. d) Aparición de personas caminando en la captura de la imagen.

pueda probarse su generalización. La figura 4.2c) muestra una imagen del mismo tipo que el de la figura 4.2b) pero esta vez con la aparición de sombras y luces provocadas por los sistemas de luz artificial fluorescente de los pasillos. Esto hace que la detección y el seguimiento de las características conlleven errores. La figura 4.2d) muestra una imagen en otro tipo de suelo con la aparición de una persona en movimiento. Esto provoca que aparezca un movimiento en el campo de visión que sea distinto (muchas veces incluso opuesto) al del movimiento del robot.

Primero se mostrarán los resultados por separado del procedimiento de odometría visual propuesto, para posteriormente comprobar como al realizar el proceso de fusión sensorial explicado anteriormente, estos resultados mejoran y se complementan.

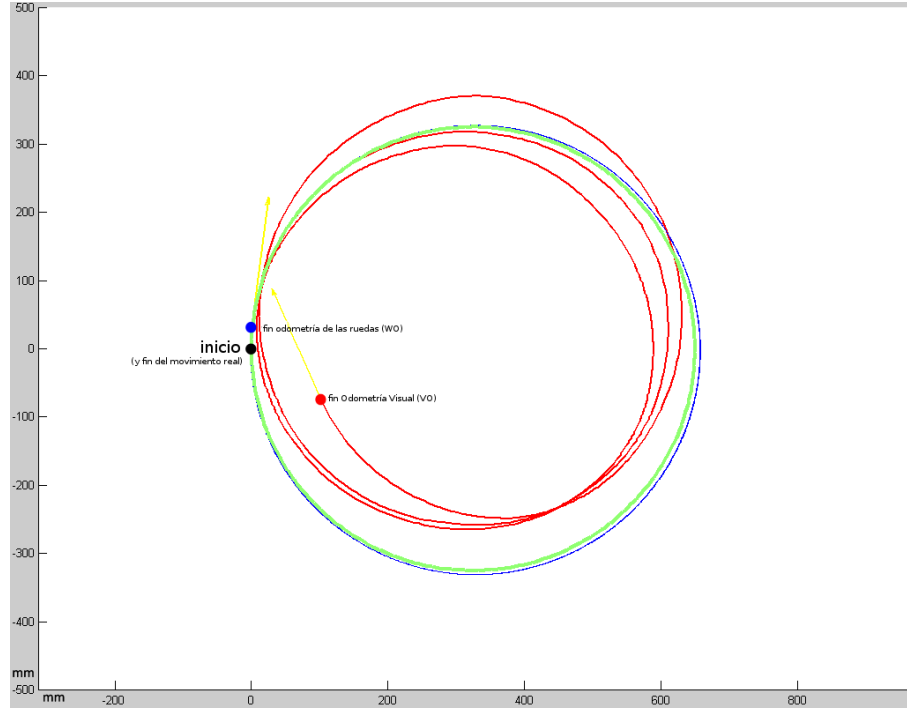


Figura 4.3: Resultados del procedimiento de odometría visual en un movimiento cíclico en condiciones idóneas. Verde: movimiento real. Azul: odometría de las ruedas (WO). Rojo: Odometría visual (VO).

Se realizaron diversas trayectorias cíclicas en los experimentos en interiores. Esto se hizo puesto que en trayectorias cíclicas es más fácil realizar las medidas del error generado. Una de los comportamientos indicados al robot fue la realización de una trayectoria circular completa durante varios ciclos, manteniendo constante el radio de giro. Es una manera precisa de comprobar la magnitud del error acumulado durante un movimiento continuo e iterativo.

La Figura 4.3 muestra el resultado de una de estas pruebas realizadas sobre movimientos circulares cíclicos con un radio de giro  $R = 325mm$ . El movimiento calculado a partir de la odometría de las ruedas está representado en azul, el movimiento calculado por la odometría visual está representado en rojo, y por último, el movimiento real está representado en color verde.

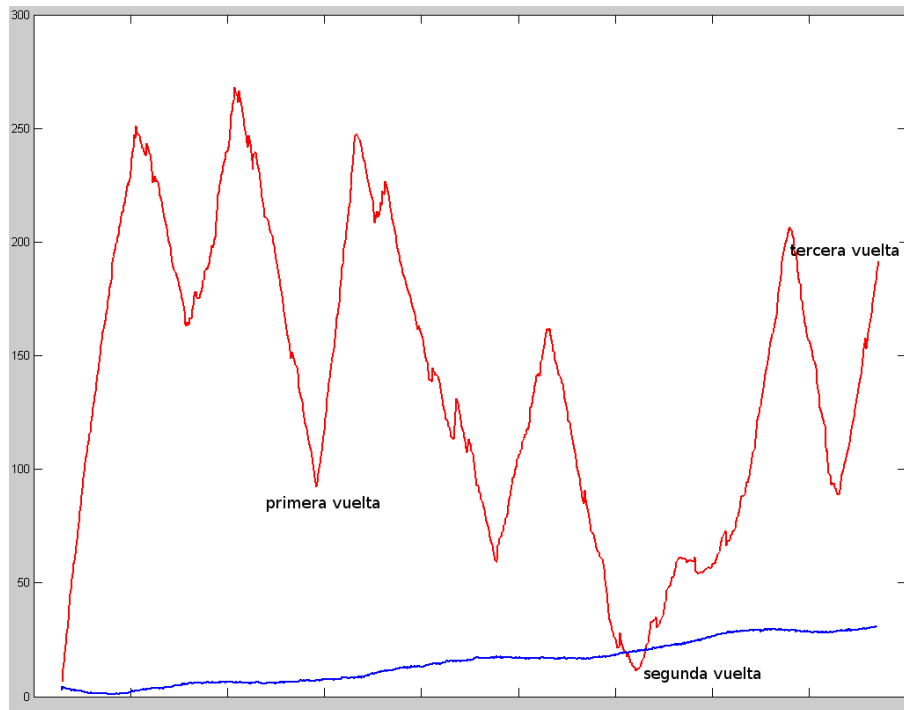


Figura 4.4: Distancia (en milímetros) entre el punto real y el punto calculado por el procedimiento de odometría visual (rojo) y la odometría de las ruedas (azul).

La figura 4.4 muestra la distancia euclídea instantánea entre los puntos de la trayectoria real y los puntos de la trayectoria calculada por la odometría visual (en rojo) y la odometría de las ruedas (en azul). En este caso, el cálculo a partir de la odometría de las ruedas tiene mejores resultados debido a las condiciones idóneas en las que se ha realizado. El suelo sobre el que se mueve el robot no tiene ninguna condición resbaladiza ni tampoco ningún tipo de obstáculo ni bache que pueda producir errores en la odometría de las ruedas. Se muestra también que el error de la odometría visual está acotado en cierta manera debido principalmente a la naturaleza cíclica del movimiento. El error de la odometría visual nunca es mayor de  $300mm$  mientras que el recorrido realizado total superó los  $6000mm$  de distancia.

Otra de las pruebas realizadas en interiores fue la realización de una navegación a lo largo de diferentes pasillos. La distancia recorrida a lo

largo de esta ruta fue cercana a los 50 metros. Durante esta navegación se produjeron cambios en la iluminación puesto que las luces fluorescentes presentes en el pasillo crean brillos cerca de los focos y se producen sombras cuando el robot se encuentra en un punto lejano respecto a los focos. Se produjeron también cambios en el tipo de suelo debido a los diferentes tipos de baldosas presentes en los diferentes pasillos. Por otra parte, existía la presencia de personas caminando cerca del robot, lo que implica no sólo apariciones de objetos extraños en la imagen sino también modificaciones en la iluminación y la aparición de sombras cuando una persona camina cerca del robot. También entre las dificultades que se presentaban durante el desarrollo de la navegación cabe destacar los ciertos baches y movimientos bruscos que se producen debido a los desniveles entre las diferentes baldosas y más concretamente en los cambios de tipo de suelo. Estas vibraciones producen imágenes desenfocadas, movimientos grandes en la imagen, etc. Todo este conjunto de anomalías hacen que el entorno no sea idóneo para la correcta ejecución del procedimiento propuesto, si bien resulta importante conocer el funcionamiento en condiciones adversas. La figura 4.5 muestra ejemplos de los diferentes tipos de imágenes que fueron capturadas durante esta prueba.

La figura 4.5a) y 4.5b) muestra un par de imágenes capturadas durante el mismo recorrido en las que se observa el cambio de iluminación notable. La primera está tomada próxima a una de las luces fluorescentes del pasillo. La segunda se encuentra en el punto medio entre dos de los focos, por lo tanto, los puntos con menor visibilidad. Las imágenes 4.5c) y 4.5d) muestran imágenes desenfocadas debido a las vibraciones que se producen durante el recorrido del robot a lo largo de los pasillos. Esto dificulta la correcta detección de características y por tanto aumenta el número de valores anómalos que existen en el cálculo de la odometría visual.

Por último, la figura 4.6 muestra el resultado de la prueba realizada. El recorrido tiene una distancia cercano a los 50 metros. Está compuesto por varios avances en línea recta y varios giros sobre sí mismo del robot. El movimiento real ocurrido está pintado en color verde y ha sido medido físicamente. El movimiento calculado a partir de la odometría de las ruedas está representado en color azul y el movimiento calculado a partir de la odometría visual de forma pura está representado en color rojo.

La figura 4.7 muestra el error en milímetros en la medida entre el punto real y el punto calculado por la odometría visual (rojo) y por la

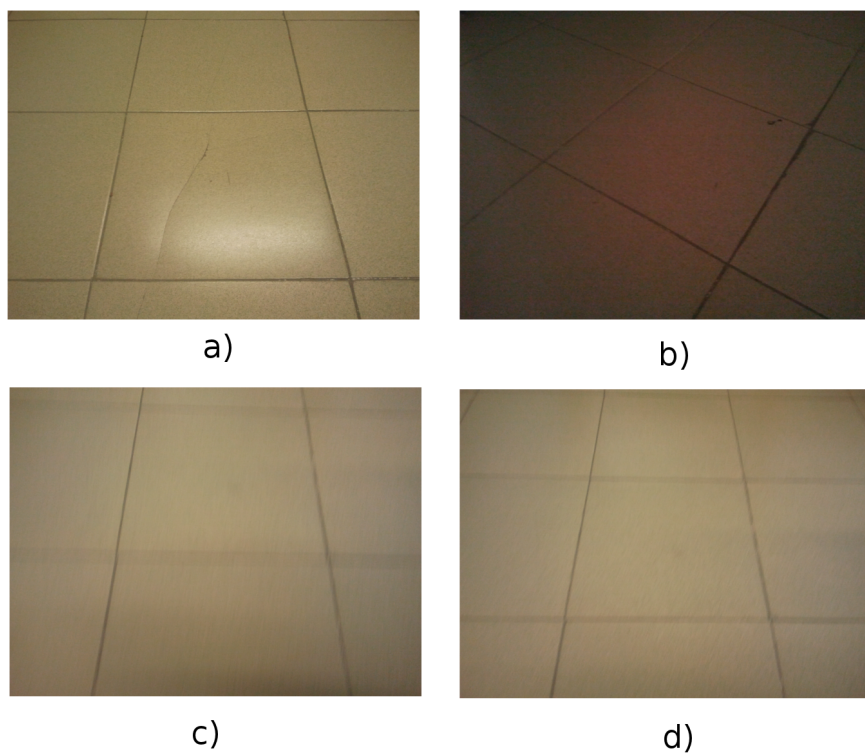


Figura 4.5: a) Imagen con mucha iluminación. b) Imagen con poca iluminación. c) Imagen desenfocada. d) Imagen desenfocada.

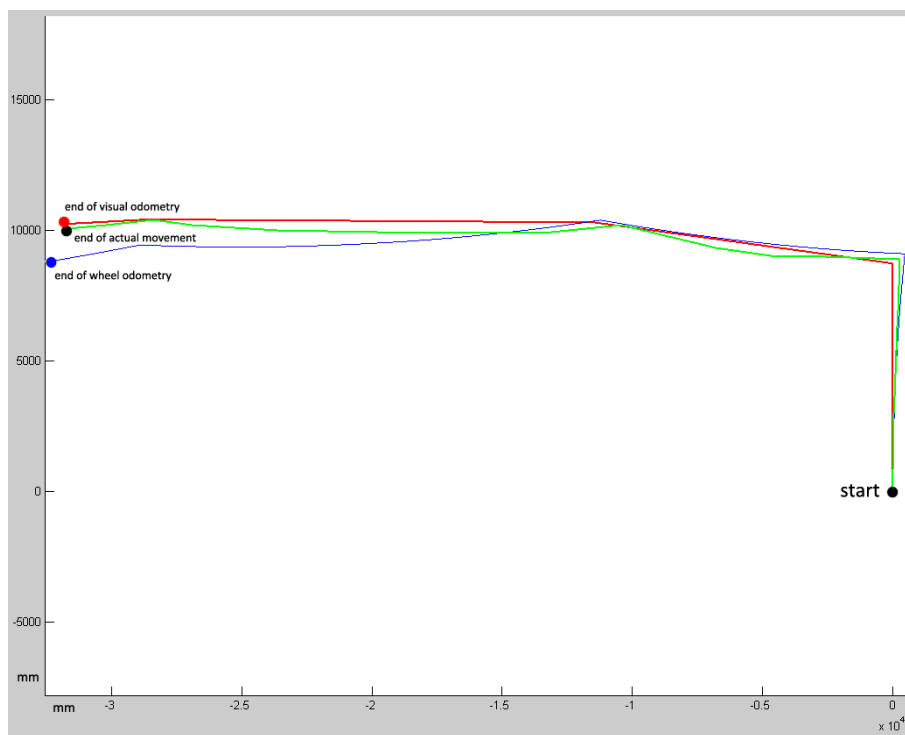


Figura 4.6: Resultados del procedimiento de cálculo de odometría visual. Verde: movimiento real. Azul: Odometría de las ruedas. Rojo: Odometría visual



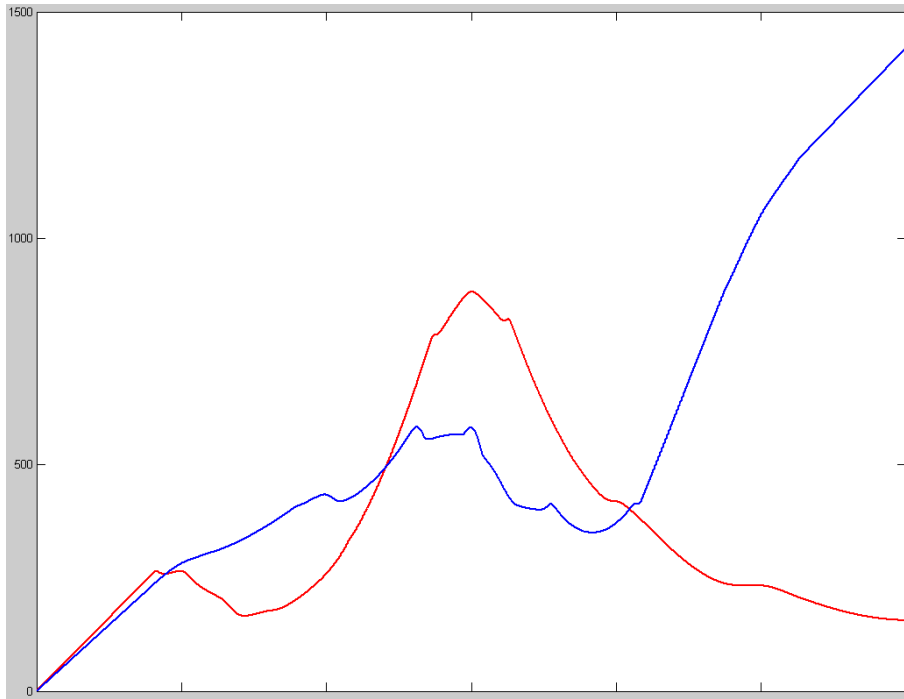


Figura 4.7: Distancia (en mm) entre el punto real y los puntos calculados por la odometría visual (rojo) y por la odometría de las ruedas (azul)

odometría de las ruedas (azul) en cada instante de tiempo durante el recorrido. El error en milímetros crece en un orden superior al lineal cuando el ángulo de giro se calcula de forma incorrecta. Como se muestra en dicha figura, en ocasiones, dos errores en el cálculo de dos giros diferentes resulta en una mejora en la medida de la distancia de error. Esto es típico de las técnicas de localización *dead-reckoning* [6], que se basan únicamente en la información obtenida en el instante de tiempo actual y en la posición del instante de tiempo anterior.

En este caso, en un recorrido de casi 50000 milímetros de distancia, el error final acumulado es de 200 milímetros. Por otra parte, el máximo error de la odometría de las ruedas resultó ser finalmente de 1500 milímetros. Este caso se desarrolla en una situación no ideal tanto para el cálculo de la odometría visual como para el cálculo de la odometría de las ruedas, dado que existen deslizamientos y existen golpes bruscos debido a los desniveles entre los diferentes tipos de baldosas. Estos desniveles

provocan errores en la medición de la odometría de las ruedas, dadas las pequeñas dimensiones del robot en sí y por tanto de las ruedas.

#### 4.2.2. Resultados en exteriores

Además de las pruebas realizadas en interiores, se realizaron diversas pruebas en exteriores para comprobar la generalización y el correcto funcionamiento del procedimiento en diferentes entornos. Se realizaron varios movimientos tanto cíclicos como en navegación común. De la misma manera que en las pruebas realizadas en interiores, se compara el resultado con la localización del robot a partir de la odometría de las ruedas.

El principal cambio respecto a las pruebas realizadas en interiores es la diferencia en las texturas del suelo. En interiores, las texturas son más simples. En exteriores, normalmente existen puntos característicos en el suelo mejor definidos. Por otra parte, respecto a la odometría de las ruedas -y más en el caso de un robot de pequeñas dimensiones como el nuestro- el entorno de exteriores es más propenso a errores debido a que los suelos son rugosos, existen baches, obstáculos, etc. Por lo tanto, la diferencia en el funcionamiento entre ambos métodos -odometría de las ruedas y odometría visual- ha de ser más evidente.

La figura 4.8 muestra diferentes imágenes tomadas durante la ejecución de los movimientos en exteriores. Se observa que las imágenes capturadas están más texturizadas que las imágenes capturadas en interiores. El proceso de detección de puntos característicos es más fácil y por lo tanto ocurren menos errores. La odometría de las ruedas es menos fiable en entornos donde el suelo no es un suelo liso, en entornos donde se producen más derrapes, deslizamientos, bloqueos de ruedas, etc. Estos dos factores hacen que la localización basada en el cálculo de la odometría visual sea más fiable en este tipo de entornos.

La figura 4.8a) muestra al robot en movimiento durante una de las pruebas realizadas. En esta figura, el robot se está aproximando a un salto. Este tipo de obstáculos y desniveles en el terreno ocasionan que la odometría de las ruedas proporcionen valores erróneos. Mientras tanto, la odometría visual es resistente a este tipo de situaciones. La figura 4.8b) muestra un suelo abrupto que conforma un entorno adverso para el cálculo de la odometría de las ruedas. La figura 4.8c) muestra con

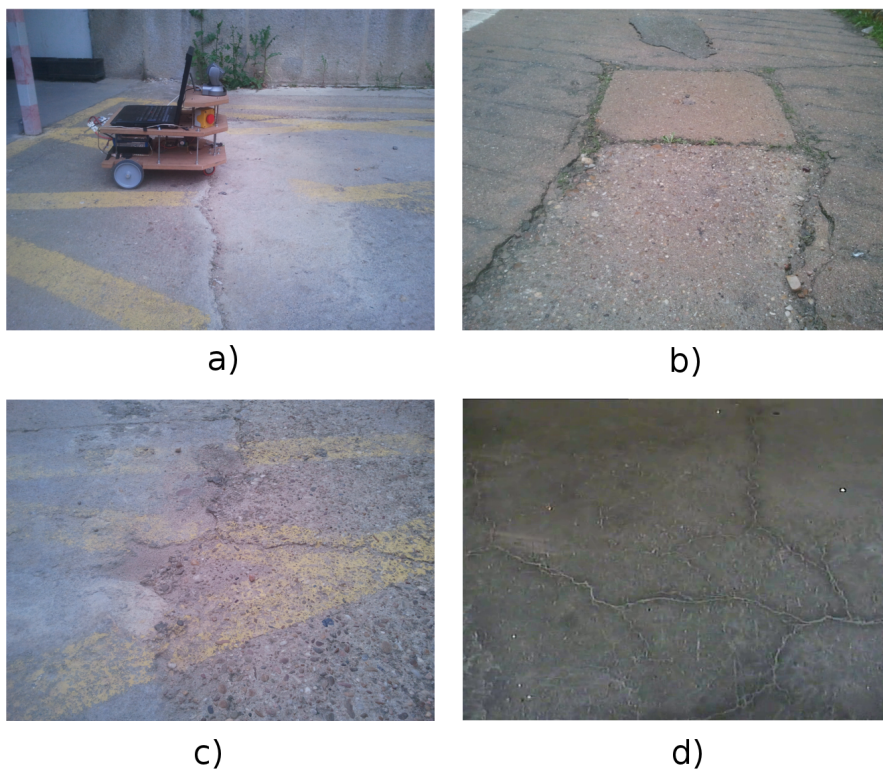


Figura 4.8: a) El robot durante la navegación. b) Suelo abrupto en exteriores. c) Un salto en el suelo. d) Suelo en exteriores menos texturizado

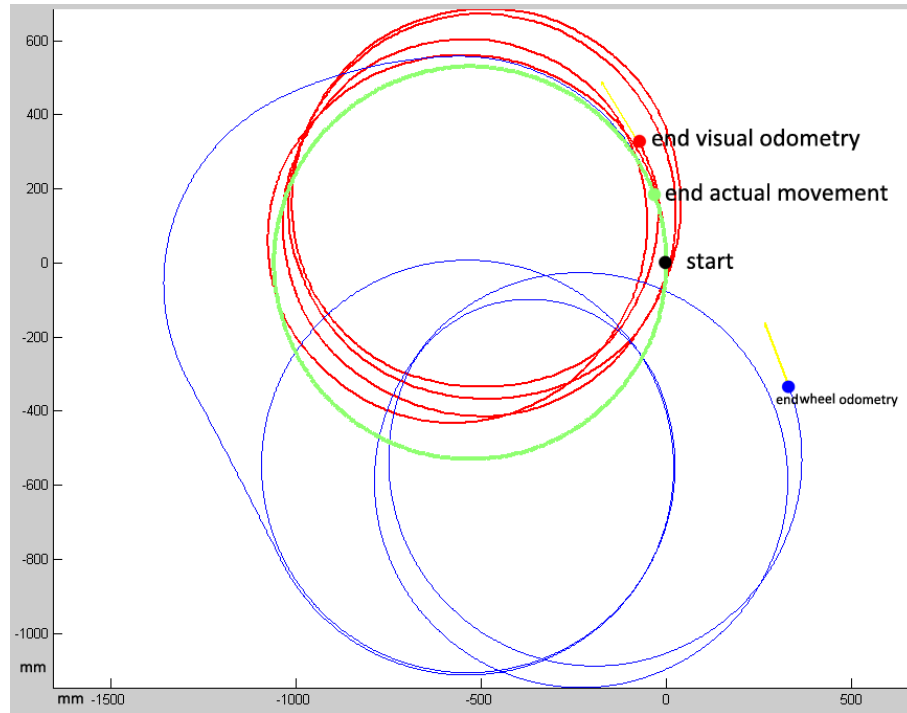


Figura 4.9: Resultados del movimiento cíclico en exteriores. Verde: movimiento real. Azul: Odometría de las ruedas. Rojo: Odometría visual

detalle el salto que se observa en la primera figura. Por último, la figura 4.8d) muestra un suelo con un texturizado diferente al de interiores, si bien también con dificultad para la obtención de puntos característicos en la imagen.

La figura 4.9 muestra los resultados del movimiento cíclico circular realizado en exteriores. El robot realizó varias circunferencias con un radio de giro constante. El movimiento calculado a partir de la odometría de las ruedas se muestra en azul, el movimiento calculado a partir de la odometría de las ruedas se muestra en color rojo y el movimiento real que se produjo está representado en verde. La ruta ordenada al robot fue la realización de cuatro giros circulares entorno al punto  $(-500, 0)$  mm. La posición final del robot contenía un error de 10 mm respecto a la posición final esperada tras las órdenes enviadas al robot. El funcionamiento de los métodos basados en el cálculo de la odometría de las ruedas empeora cuando el robot realiza movimientos en el exterior debido a la mayor

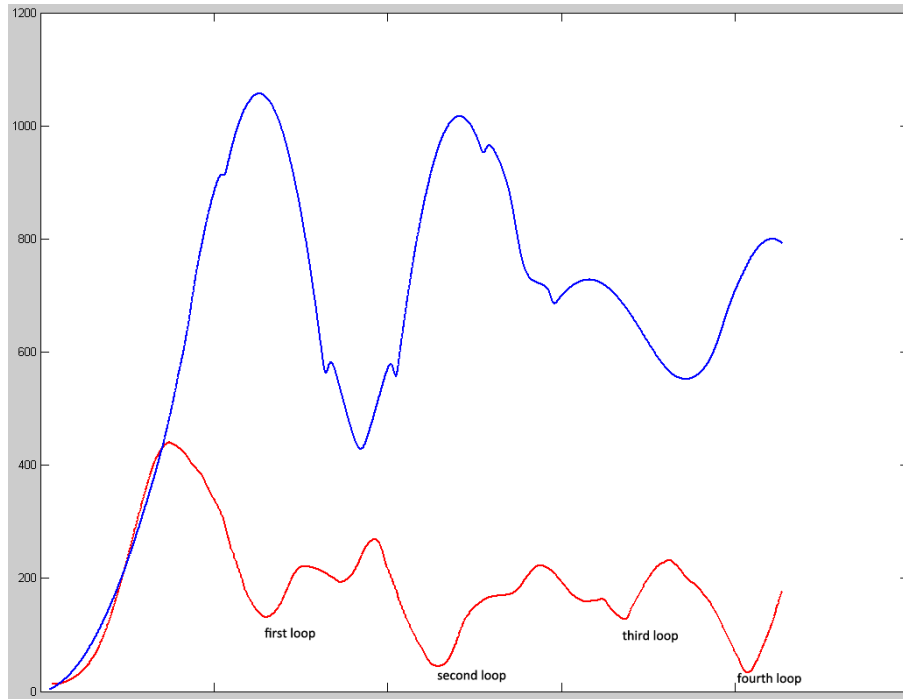


Figura 4.10: Distancia (en mm) entre el punto real y los puntos calculados por la odometría visual (rojo) y por la odometría de las ruedas (azul)

rugosidad del suelo, los derrapes de las ruedas, los obstáculos que se encuentran en el suelo que pueden hacer que las ruedas patinen o incluso que se queden bloqueadas, etc. En el caso de que la rueda se quede atascada o derrape, la odometría visual se mantendrá constante. Sin embargo, la odometría de las ruedas introducirá errores en la localización asumiendo que se está produciendo un movimiento cuando no es así.

La figura 4.10 muestra que el error de la odometría visual es más o menos constante a lo largo del tiempo mientras que el error de la odometría de las ruedas aumenta cuando las ruedas tienen problemas con los obstáculos que existen en el suelo.

La figura 4.11 muestra el resultado de un recorrido realizado en exteriores. Como en las figuras de resultados anteriores, se muestran las trayectorias calculadas en base al cálculo de la odometría visual (en rojo), en base a la odometría de las ruedas (azul) y el movimiento real que se ha ordenado al robot (verde). El robot se atascó varias veces durante

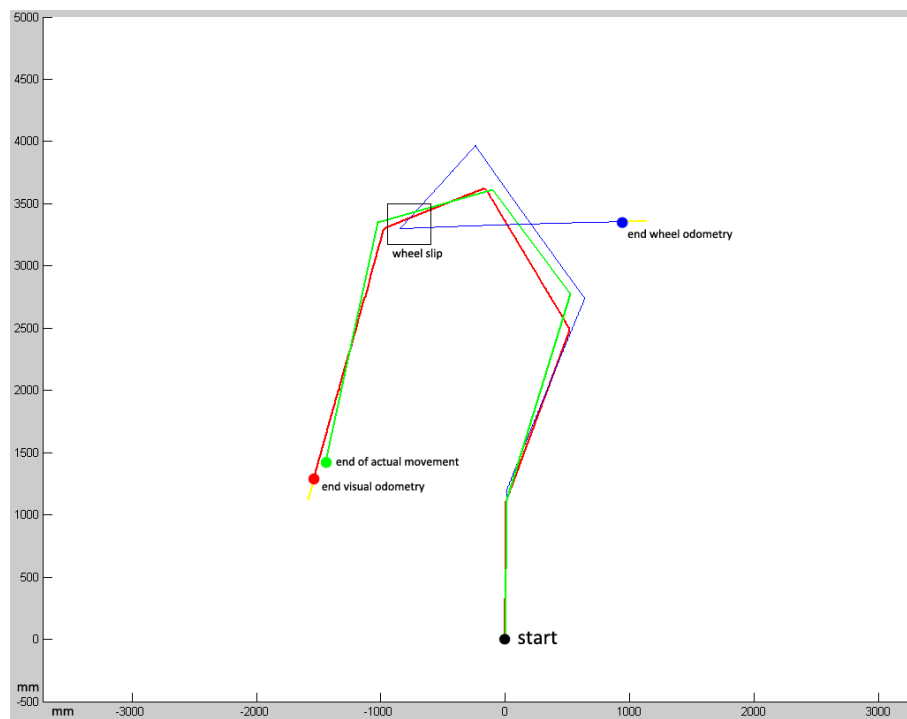


Figura 4.11: Recorrido realizado en exteriores. Verde: movimiento real. Azul: Odometría de las ruedas. Rojo: Odometría visual

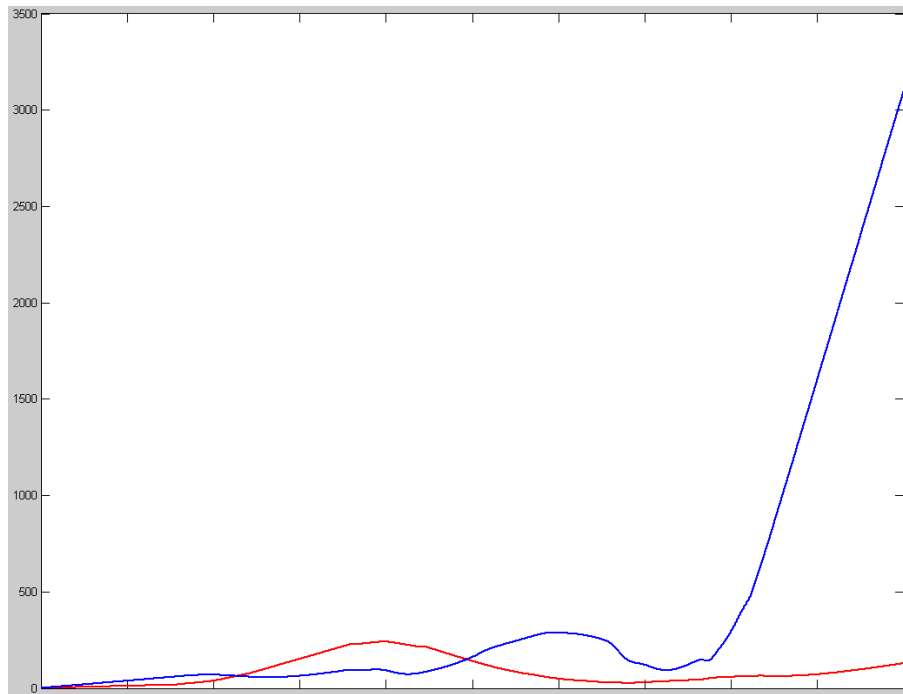


Figura 4.12: Distancia (en mm) entre el punto real y los puntos calculados por la odometría visual (rojo) y por la odometría de las ruedas (azul)

la realización del movimiento. En la figura 4.11 se muestra el punto en el que la rueda se queda atascada y que afecta posteriormente a toda la localización basada en el cálculo de la odometría de las ruedas. En condiciones no ideales, la localización basada únicamente en el cálculo de la odometría de las ruedas es poco robusta y los errores pueden ser muy grandes.

La figura 4.12 muestra como el error de la odometría de las ruedas y el de la odometría visual es similar hasta el momento en el que la rueda queda atascada. En ese momento el error de la odometría de las ruedas crece indefinidamente mientras que el error de la odometría visual se mantiene constante.

### 4.3. Adecuación a la ejecución en tiempo real

El procedimiento presentado tiene como objetivo la ejecución en un entorno de tiempo real durante el movimiento del robot. De esta manera, podrá ser utilizado por el robot como parte del sistema de localización que favorezca la realización de una navegación autónoma.

Es por ello que ha sido necesario priorizar los parámetros de coste temporal sobre otras características a la hora de elegir los procesos que se iban realizar durante el procedimiento de cálculo de la odometría visual. Estos procesos se han explicado con detalle y evaluado en las secciones anteriores.

Se ha realizado un estudio sobre los tiempos de ejecución de los diferentes procesos que forman parte del procedimiento del cálculo de la odometría visual. La muestra sobre la que se realiza el estudio consiste en más de  $10^5$  ejecuciones del procedimiento entre dos imágenes consecutivas. La tabla 4.1 muestra la media de los tiempos de ejecución de los diferentes procesos que componen el procedimiento completo de cálculo de odometría visual. Para que el estudio pudiera ser generalizado y se pudiera demostrar la adecuación del procedimiento en un entorno de tiempo real, las pruebas se realizaron utilizando tres entornos de ejecución domésticos distintos.

El primero entorno consta de un procesador Intel Core 2 Duo 3.0 GHz, que es el portátil que se encuentra a bordo del robot en la actualidad. El segundo entorno de ejecución consta de un procesador Intel i7-2600 3.40 GHz. Por último, el tercer entorno de ejecución es un procesador Mac Pro 2 Quad-Core Intel Xeon 2.8 GHz.

La ejecución del procedimiento se realiza en una aplicación multihilo. La captura de las imágenes se realiza siempre en un hilo diferente al que realiza los cálculos de odometría visual. Esto implica que el tiempo que se necesita para capturar la imagen no se añade realmente al tiempo total de ejecución puesto que mientras se está capturando la nueva imagen, al mismo tiempo se están realizando los procesos necesarios para el cálculo de la odometría visual entre las dos imágenes anteriores.

De esta forma, si tomamos los tiempos de ejecución del ordenador doméstico que está actualmente sobre el robot, el procedimiento de cálculo de odometría visual entre dos imágenes tiene un coste temporal de 40 milisegundos. Esto permite que se pueda utilizar la máxima tasa de cap-



	Ent 1	Ent 2	Ent 3
Captura de imagen	40(25fps)	40(25fps)	40(25fps)
Detección características	18,41	9,90	15,18
Seguimiento características	11,16	5,67	9,95
2D a 3D	1,17	0,97	1,01
Cálculo CIR	1,52	1,62	2,54
Eliminar valores anómalos	0,56	0,41	0,86
Clasificador borroso	0,71	1,12	1,51
Cálculo de la nueva posición	0,70	0,14	0,52
TOTAL	34,03	19,83	31,57

**Tabla:** 4.1: Tiempo de ejecución del procedimiento VO (en milisegundos)

tura de la cámara que se encuentra actualmente en el robot, esto es, 25 imágenes por segundo. Por ello, se puede afirmar, que en base a parámetros temporales, el procedimiento puede ser utilizado como parte del sistema de localización del robot durante una navegación en tiempo real.

#### 4.4. Caso práctico: Experiencia de fusión sensorial

El procedimiento de odometría visual propuesto es un procedimiento puramente exteroceptivo. La información necesaria para el correcto resultado del procedimiento se toma utilizando únicamente sensores externos al robot. Sin embargo, como caso práctico se ha planteado también la utilización de métodos internos que puedan proporcionar mayor robustez al sistema en caso de necesidad. De esta manera se busca minimizar el error que se produce en el proceso del cálculo de la localización de un robot. En este caso, los sensores que se utilizan como fuente de información son los encoders de las ruedas del robot y la cámara del robot. Ambos sensores proporcionan información relativa al movimiento del robot partiendo de la localización del robot en el instante de tiempo inmediatamente anterior y el movimiento realizado entre ambos instantes de tiempo.

Ambas medidas contienen un error que aumenta con el paso del tiempo, puesto que ambos cálculos dependen de los estados anteriores y por tanto, no son globales. Es por ello que se hace si cabe más necesario la fusión de la información de varios sensores para así reducir el error en el cálculo del movimiento realizado por el robot entre dos instantes de tiempo consecutivos.

En el capítulo 3, se explicaba el procedimiento de cálculo de la odometría visual a partir de las imágenes capturadas por la cámara que se encuentra en el robot. Este procedimiento nos proporciona el movimiento que ha tenido lugar entre dos instantes de tiempo a partir de dos imágenes capturadas en esos dos instantes de tiempo respectivamente. Una vez conocido el movimiento que ha tenido lugar, se puede calcular la nueva localización del robot a partir del estado del robot en el instante inicial.

De la misma manera, se puede calcular el movimiento realizado por el robot entre dos instantes tiempo a partir de los encoders de las ruedas. De nuevo, una vez conocido el movimiento que ha tenido lugar y conocida también la posición en el instante inicial, se puede calcular la nueva localización del robot.

El objetivo de este capítulo es explicar el procedimiento realizado para fusionar ambas medidas de forma que el error quede minimizado. Ambos sistemas de localización son sistemas sensibles a errores. Un error en el sistema de localización puede hacer que la tarea encomendada a un robot se realice de forma incorrecta. El uso de técnicas de localización basadas en la odometría de las ruedas es común en la robótica, bien sea como único método de localización o como uno de los diferentes métodos utilizados para el cálculo del estado de un robot.

Sin embargo, la odometría de las ruedas es dependiente del entorno de trabajo puesto que los encoders de las ruedas son sensores propioceptivos. Así, por ejemplo, en un suelo liso y sin obstáculos, la odometría de las ruedas resulta ser un sistema bastante preciso. Un ejemplo de ello lo tenemos en los robots que trabajan en entornos de oficinas, con suelos nada rugosos y sin obstáculos. Pero por otra parte, en suelos rugosos, con ondulaciones, obstáculos o resbaladizos, la localización a partir de la odometría de las ruedas conlleva muchos errores. Por ejemplo, un robot que realice operaciones en el exterior de un edificio puede encontrar múltiples problemas, como son piedras en el suelo, barro, etc. Este tipo

de problemas inhabilita el uso de la odometría de las ruedas para la localización de un robot.

La odometría visual proporciona información fiable en diferentes entornos, puesto que su información procede de un sensor exteroceptivo como es la cámara. Sin embargo, es sensible a errores puesto que la aparición de objetos móviles en el campo de visión puede provocar que se observe un movimiento en la imagen que no ha sido realizado por el robot y, sin embargo, se tome como tal. Si bien en el capítulo anterior se ha explicado los procesos de filtrado que se llevan a cabo para evitar estos errores, en ocasiones es posible que la localización basada en odometría visual no proporcione resultados correctos, por ejemplo en ambientes donde existan muchos objetos móviles en la imagen, o sencillamente se produzcan problemas con la iluminación.

Es por ello, que la fusión de ambas odometrías resulta en una medida más fiable donde las debilidades de una se compensan con las fortalezas de la otra. De esta manera se logra un valor de odometría más robusto que puede ser utilizado en diversas situaciones. La fusión de los datos procedentes de ambos sensores odométricos se realiza utilizando el Filtro de Kalman Extendido, detallado en sus aspectos teóricos en el Anexo A.

#### 4.4.1. Implementación del EKF

De manera similar al filtro de Kalman lineal, el algoritmo del EKF se basa en un ciclo de predicción-corrección, con la diferencia de que se linealizan las funciones no lineales mediante una serie de Taylor de primer orden. A continuación se muestran las ecuaciones sin su desarrollo matemático.

**PREDICCIÓN:** Dada la estimación del estado del sistema y su matriz de covarianza  $(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$  en el instante de tiempo  $k$ , la predicción  $(\hat{\mathbf{x}}_{k+1|k}, \mathbf{P}_{k+1|k})$  en el instante de tiempo  $k + 1$  se calcula como:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k) \quad (4.1)$$

$$\mathbf{P}_{k+1|k} = \nabla_x \mathbf{f} \cdot \mathbf{P}_{k|k} \cdot \nabla_x \mathbf{f}^T + \nabla_u \mathbf{f} \cdot \mathbf{Q}_k \cdot \nabla_u \mathbf{f}^T \quad (4.2)$$

La incertidumbre asociada a la posición en el instante de tiempo  $k + 1$  ( $\mathbf{P}_{k+1|k}$ ) depende de dos variables:

1. La matriz de covarianza  $\mathbf{P}_{k|k}$ , que representa la incertidumbre asociada a la posición en el instante de tiempo anterior.
2. La matriz de covarianza  $\mathbf{Q}_k$ , que define la incertidumbre relacionada con la entrada de control, es decir, con el desplazamiento medido por los encoders.

En la ecuación (4.2) aparecen dos Jacobianos que preceden y suceden a cada una de estas dos variables, y su función es linealizarlas. Por una parte, el Jacobiano de la función de transición de estados ( $\nabla_x \mathbf{f}$ ) se debe evaluar en el estado  $\mathbf{x} = \hat{\mathbf{x}}_{k|k}$  y se calcula como:

$$\nabla_x \mathbf{f} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x} & \frac{\partial \mathbf{f}}{\partial y} & \frac{\partial \mathbf{f}}{\partial \theta} & \frac{\partial \mathbf{f}}{\partial \dot{x}} & \frac{\partial \mathbf{f}}{\partial \dot{y}} & \frac{\partial \mathbf{f}}{\partial \dot{\theta}} \end{bmatrix}$$

$$\nabla_x \mathbf{f} = \begin{bmatrix} 1 & 0 & \Delta S_k \cdot \cos\left(\theta_k + \frac{\Delta\theta_k}{2}\right) & 0 & 0 & \frac{\Delta S_k \cdot \cos\left(\frac{\theta_k + \Delta\theta_k}{2}\right)}{2} \\ 0 & 1 & \Delta S_k \cdot \sin\left(\theta_k + \frac{\Delta\theta_k}{2}\right) & 0 & 0 & -\frac{\Delta S_k \cdot \sin\left(\frac{\theta_k + \Delta\theta_k}{2}\right)}{2} \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \frac{\Delta S_k \cdot \cos\left(\frac{\Delta\theta_k}{2}\right)}{2} \\ 0 & 0 & 0 & 0 & 0 & -\frac{\Delta S_k \cdot \sin\left(\frac{\Delta\theta_k}{2}\right)}{2} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

Por otra parte, el Jacobiano de la entrada de control ( $\nabla_u \mathbf{f}$ ) se evalúa igualmente en el estado  $\mathbf{x} = \hat{\mathbf{x}}_{k|k}$ , y se calcula como:

$$\nabla_u \mathbf{f} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \Delta S_i} & \frac{\partial \mathbf{f}}{\partial \Delta S_d} \end{bmatrix}$$

$$\nabla_u \mathbf{f} = \begin{bmatrix} (t_1 + t_2) & (t_1 - t_2) \\ (t_3 - t_4) & (t_3 + t_4) \\ t_5 & -t_5 \\ (t_6 + t_7) & (t_6 - t_7) \\ (t_8 - t_9) & (t_8 + t_9) \\ t_5 & -t_5 \end{bmatrix}$$

$$\begin{aligned} t_1 &= \frac{1}{2} \cdot \sin \left( \theta_k + \frac{\Delta \theta_k}{2} \right) \\ t_2 &= \frac{1}{2b} \cdot \Delta S_k \cdot \cos \left( \theta_k + \frac{\Delta \theta_k}{2} \right) \\ t_3 &= \frac{1}{2} \cdot \cos \left( \theta_k + \frac{\Delta \theta_k}{2} \right) \\ t_4 &= \frac{1}{2b} \cdot \Delta S_k \cdot \sin \left( \theta_k + \frac{\Delta \theta_k}{2} \right) \\ t_5 &= \frac{1}{b} \\ t_6 &= \frac{1}{2} \cdot \sin \left( \frac{\Delta \theta_k}{2} \right) \\ t_7 &= \frac{1}{2b} \cdot \Delta S_k \cdot \cos \left( \frac{\Delta \theta_k}{2} \right) \\ t_8 &= \frac{1}{2} \cdot \cos \left( \frac{\Delta \theta_k}{2} \right) \\ t_9 &= \frac{1}{2b} \cdot \Delta S_k \cdot \sin \left( \frac{\Delta \theta_k}{2} \right) \end{aligned} \quad (4.4)$$

donde  $b$  es la distancia entre las ruedas motrices.

**CORRECCIÓN:** En el instante de tiempo  $k+1$  se realiza la observación  $\mathbf{z}_{k+1}$ , y utilizando la predicción del estado calculada en la fase anterior, se corrige la estimación del estado de acuerdo a las ecuaciones:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1} \cdot \tilde{\mathbf{z}}_{k+1} \quad (4.5)$$

$$\mathbf{P}_{k+1|k+1} = [\mathbf{I} - \mathbf{K}_{k+1} \cdot \nabla_x \mathbf{h}] \cdot \mathbf{P}_{k+1|k} \quad (4.6)$$

donde (4.5) es la corrección de la estimación del estado, y (4.6) es la corrección de la covarianza del estado, y los términos:

$$\tilde{\mathbf{z}}_{k+1} = \mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k}) \quad (4.7)$$

$$\mathbf{S}_{k+1} = \nabla_x \mathbf{h} \cdot \mathbf{P}_{k+1|k} \cdot \nabla_x \mathbf{h}^T + \mathbf{R}_{k+1} \quad (4.8)$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \cdot \nabla_x \mathbf{h}^T \cdot \mathbf{S}_{k+1}^{-1} \quad (4.9)$$

representan la innovación (4.7), la covarianza de la innovación (4.8), y la ganancia subóptima (o quasi-óptima) de Kalman (4.9).

La función de observación se linealiza mediante el Jacobiano ( $\nabla_x \mathbf{h}$ ), evaluado utilizando la última predicción del estado  $\mathbf{x} = \hat{\mathbf{x}}_{k+1|k}$ :

$$\begin{aligned} \nabla_x \mathbf{h} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & t_{10} & t_{11} & 0 \end{bmatrix} \\ t_{10} &= \frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ t_{11} &= \frac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \end{aligned} \quad (4.10)$$

#### 4.4.1.1. Compensación heurística de los errores

Dado que el EKF se obtiene mediante una linealización basada en una serie de Taylor de primer orden, se están introduciendo errores en los cálculos, debido principalmente a que:

- Los términos de segundo orden o mayores de la serie de Taylor no se están teniendo en cuenta.
- Las evaluaciones de los Jacobianos se están haciendo utilizando valores estimados del estado en lugar de utilizar los valores reales, dado que éstos no están disponibles.

Para compensar estos errores, que pueden llevar a la divergencia del filtro, se pueden utilizar los siguientes métodos heurísticos:

- Añadir ruido artificial o pseudo-ruido al modelo del sistema para compensar los errores en la predicción del estado. Estos se puede lograr utilizando una matriz de covarianza del sistema modificada  $\mathbf{Q}_k^m$  en la ecuación de predicción de la covarianza (4.2):

$$\mathbf{Q}_k^m = \mathbf{Q}_k^p + \mathbf{Q}_k \geq \mathbf{Q}_k \quad (4.11)$$

donde  $\mathbf{Q}_k$  es una matriz de covarianza con ruido artificial.

- Multiplicar la covarianza del estado por un escalar  $\phi > 1$  en cada tiempo de muestreo:

$$\mathbf{P}_{k+1|k}^\phi = \phi \cdot \mathbf{P}_{k+1|k} \quad (4.12)$$

y utilizar esta matriz modificada en la ecuación de predicción de la covarianza (4.2). La multiplicación de la matriz de covarianza por un escalar mayor que la unidad es equivalente a un filtro que pierde la memoria de lo sucedido, ya que es como si las últimas actualizaciones de la covarianza se hubiesen perdido, ya que la incertidumbre se hace mayor.

Mediante cualquiera de estas dos técnicas se consigue que la matriz de covarianza cubra los errores cometidos al linealizar un sistema no lineal. Al incrementar el error en la matriz de covarianza del estado ( $\mathbf{Q}_k$ ) se consigue que la ganancia del filtro sea mayor, dándole mayor peso a las observaciones más recientes. Si por el contrario se incrementa el error en la matriz de covarianza de las observaciones ( $\mathbf{R}_k$ ), se obtendría el efecto opuesto: bajar la ganancia del filtro y ponderar en menor medida las nuevas observaciones.

#### 4.4.2. Resultados de la Fusión Sensorial

Una vez realizadas diversas pruebas para comprobar el correcto funcionamiento del procedimiento de cálculo de odometría visual propuesto, se han realizado diversas pruebas en las que se incorporan otras mediciones tomadas a partir de otros sensores del robot. En este caso, se toma la medida de los *encoders* de las ruedas del robot.

El sistema de movimiento del robot consta de dos ruedas motrices traseras y una rueda loca delantera, siguiendo por tanto una configuración diferencial. Las dos ruedas traseras poseen un par de encoders lineales a partir de los que se puede calcular el movimiento realizado por el robot.

De esta manera, se tienen dos fuentes de medición: la odometría calculada a partir del movimiento de las ruedas del robot y la odometría calculada a partir de las imágenes capturadas por el sistema de visión que se encuentra sobre el robot.

Con la fusión de ambos sensores se pretende la obtención de un sensor odométrico fiable en diversos entornos y diversas circunstancias. La

odometría de las ruedas ha demostrado ser un sistema de localización fiable en entornos concretos e ideales. Los sistemas de localización basados en la odometría de las ruedas proporcionan resultados fiables cuando la navegación se produce sobre suelos lisos, no resbaladizos, sin cambios abruptos (saltos o baches), que puedan producir bloqueos o patinazos en las ruedas. Es un sistema que toma la información de sensores internos del robot.

La localización a partir de la odometría de las ruedas es un sistema de localización en el que el error aumenta continuamente a lo largo del tiempo. No existe ningún método de corrección. Por tanto, este tipo de sistemas de localización suelen ir acompañados de otros sistemas exteroceptivos que disminuyen el error. Estos sistemas pueden basarse en balizas o en elementos reconocibles y diferenciados en el entorno, como puede ser la localización Wi-Fi, el uso del reconocimiento de elementos en un edificio (puertas, pasillos, salas, etc).

Por otra parte, lo que se propone en esta tesis es la incorporación de otro sistema odométrico exteroceptivo que mitigue los errores que se pueden producir en la odometría de las ruedas. La odometría de las ruedas es muy sensible a errores. Un cambio abrupto en el suelo o incluso el bloqueo de una rueda pueden provocar que el error en el cálculo de la posición del robot sufra un error inmenso. De la misma manera, el hecho de que una rueda patine puede provocar que ese error se propague a lo largo de la navegación llegando incluso a resultar inútil la localización calculada.

En el apartado anterior, se ha demostrado que la odometría visual tiene un comportamiento fiable en diferentes entornos, incluido aquéllos en los que la odometría de las ruedas falla estrepitosamente. Por ejemplo, en el caso de un derrape de una sola rueda, el resultado de la odometría de las ruedas será que se ha producido un giro sobre la rueda que no ha derrapado, puesto que la rueda que ha derrapado se habrá movido más espacio que la otra rueda. Un cambio en la orientación del robot provocará que ese error se propague con el paso del tiempo disminuyendo la fiabilidad de la posición calculada.

De la misma manera, el bloqueo de ambas ruedas, debido por ejemplo a un bache en el suelo, provocará que el sistema de odometría de las ruedas proporcione como resultado que el robot está realizando un movimiento en línea recta, cuando en realidad el robot se encuentra en un



estado de no movimiento (con ambas ruedas bloqueadas, pero girando).

En ambos casos, la odometría visual no cae en dicho error. Al tomar como información de entrada las imágenes capturadas por el sistema de visión, se puede considerar el sistema de odometría visual como un sistema exteroceptivo y al sistema de visión como un sensor externo del robot. En el caso de un derrape de una única rueda durante un avance en línea recta, efectivamente las imágenes mostrarán que el robot continúa avanzando de manera normal, puesto que las imágenes no mostrarán giro alguno.

En el caso de que ocurra un bloqueo de ambas ruedas (debido a algún bache o elemento en el suelo), el sistema de localización basado en la odometría visual dará como resultado un movimiento nulo. Las imágenes capturadas por el sistema de visión mostrarán que efectivamente el robot no se está moviendo.

De esta manera, parece lógico pensar que si ambos sistemas pueden complementarse, el resultado de realizar una fusión sensorial entre los *encoders* de las ruedas y el sistema de visión será un sistema odométrico más robusto y fiable.

En este apartado, se mostrarán a modo de aplicación, dos ejemplos de navegación expuestos con anterioridad.

El primero de ellos se mostró en la Figura 4.6. El entorno de navegación en este ejemplo fueron los pasillos de la Facultad de Ciencias de la Universidad de Salamanca. El entorno es un entorno ideal, en un suelo con baldosas y sin ningún tipo de obstáculo, bache, ni elemento que pueda provocar un deslizamiento de las ruedas. El resultado en la localización a partir de los datos de la odometría de las ruedas resultan ser fiables. De la misma manera, la localización utilizando la odometría visual también resulta fiable, a pesar de los cambios de iluminación que se producen en las imágenes debido a las luces fluorescentes que existen en el pasillo.

La Figura 4.13 muestra los diferentes resultados de localización obtenidos durante la navegación del robot. En azul se muestra el movimiento realizado basándose en la odometría de las ruedas. En color rojo se muestra la navegación realizada basándose en la odometría visual. En verde se muestra la trayectoria real que ha seguido el robot.

Por último, en negro se muestra el resultado de la implementación de

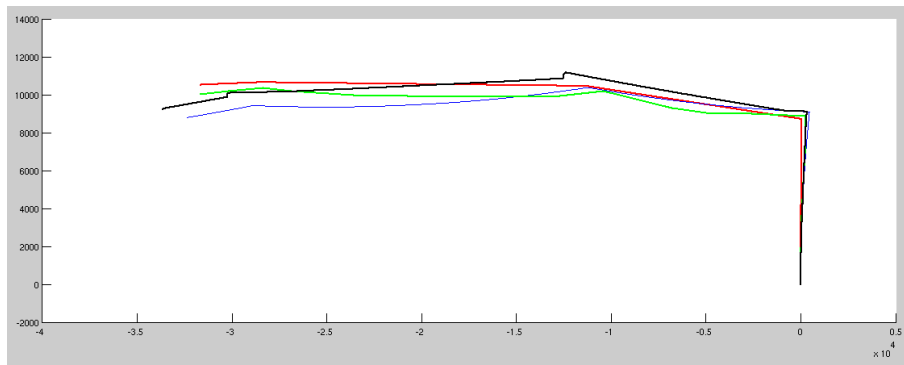


Figura 4.13: Resultados de la fusión sensorial. Azul: Odometría de las ruedas. Rojo: Odometría visual. Negro: Fusión sensorial. Verde: Trayectoria real

un filtro de Kalman extendido (EKF) tomando como medida del sistema los datos procedentes de la odometría de las ruedas y como medida de observación los datos procedentes de la odometría visual. Se observa claramente que el resultado de la fusión sensorial produce un resultado igualmente fiable, con un error admisible de un metro tras un avance de cerca de 50 metros.

El segundo ejemplo que se muestra en este apartado será el mismo que el mostrado en la Figura 4.11. Durante el recorrido realizado se produce el derrape de una de las ruedas del robot durante un cierto tiempo. Esto provoca que la localización a partir de la odometría de las ruedas devuelva un resultado completamente erróneo. Durante el período en el que una de las ruedas está derrapando, la odometría basada en el movimiento de las ruedas considera que el robot está realizando un giro (en este caso hacia la izquierda). Sin embargo, la odometría visual devuelve que el robot no está en movimiento, puesto que las imágenes capturadas por el sistema de visión no muestran movimiento alguno.

De esta manera, un sistema de fusión sensorial entre ambas odometrías mitigará el error que introduce la odometría de las ruedas. En la Figura 4.14 se muestra el resultado de la localización durante la navegación del robot que se tiene como resultado de la odometría de las ruedas en azul. En color rojo, se muestra la navegación en base a la odometría visual y en verde la trayectoria real del robot.

Por su parte, en negro se muestra el resultado de la fusión senso-

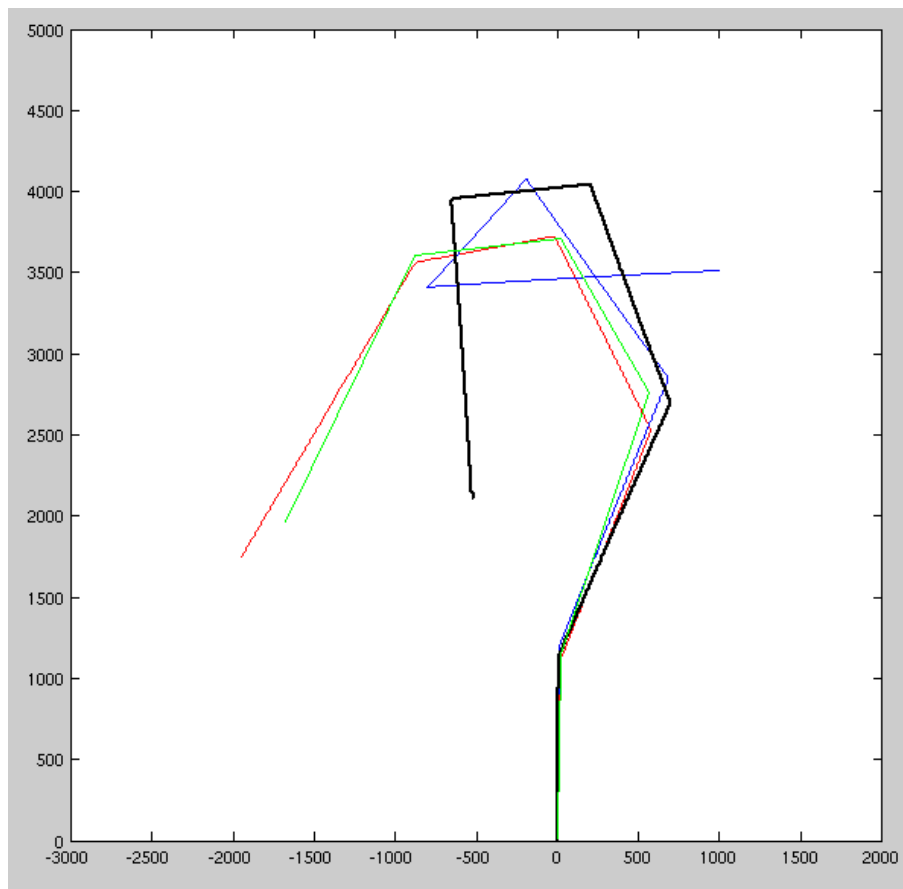


Figura 4.14: Resultados de la fusión sensorial. Azul: Odometría de las ruedas. Rojo: Odometría visual. Negro: Fusión sensorial. Verde: Trayectoria real

rial entre ambas odometrías. Se observa claramente que el error que se introduce al derrapar una de las ruedas queda mitigado en parte, disminuyendo claramente el error final en la localización del robot. Se observa también que una vez que el derrape finaliza (y el robot comienza a avanzar en línea recta) el sistema considera correctamente dicho avance, a pesar de que las orientaciones de ambos sistemas odométricos no son la misma (incluso casi opuestas).

De esta manera, se demuestra de forma aplicada que con un sistema de fusión sensorial sencillo se puede crear un sistema odométrico más robusto y más fiable para diferentes entornos. La odometría de las ruedas es muy fiable en entornos de suelos lisos, no resbaladizos y sin obstáculos. La odometría visual es muy fiable en entornos donde la iluminación no sufre excesivos cambios y donde los movimientos ajenos al entorno son pequeños. Esto significa que en un entorno de iluminación variable o con muchos elementos móviles en la escena, los errores introducidos por la odometría visual serán mitigados por la odometría de las ruedas. De la misma manera, los errores introducidos por la odometría de las ruedas en caso de bloqueos o derrapes serán mitigados por el sistema de odometría visual.

Por supuesto, existen diferentes métodos de fusión sensorial que se pueden aplicar con la misma intención de obtener un sistema de odometría robusto y fiable en cualquier entorno. La aplicación y estudio de estos métodos más complejos es objeto de próximos trabajos.



## Capítulo 5

# Conclusiones

En este capítulo se describe cómo se han conseguido los objetivos propuestos en el inicio del trabajo de investigación, cómo se ha evaluado y validado la hipótesis inicial: *la propuesta e implementación de un sistema completo de odometría visual monocular bajo restricciones de tiempo real*.

### 5.1. Conclusiones y principales contribuciones

El trabajo de investigación que se presenta en este documento proporciona una contribución novedosa en el campo de la localización basada en la odometría visual monocular. El sistema propuesto es hábil para trabajar con una frecuencia de muestreo de imágenes aceptable bajo restricciones de tiempo real, aceptando, sin ocasionar errores graves, oclusiones y movimientos en el campo de visión. Todo el sistema está basado en componentes domésticos de bajo coste.

Todo el proceso desde la propuesta de la hipótesis hasta la obtención y validación de los resultados así como la metodología utilizada para la validación de la hipótesis inicial se describen a lo largo de este documento.

Las principales contribuciones de este trabajo de investigación se presentan a continuación:

**Estudio de los trabajos relacionados, tecnologías y metodologías**

Al comienzo de esta investigación, se realizó un estudio de las diferentes estrategias realizadas hasta el momento en problemas de localización utilizando técnicas de odometría visual. Este estudio se convierte en un compendio completo con la literatura imprescindible y necesaria a la hora de afrontar un problema de investigación en un área concreta para nuevos investigadores. Además, se han incorporado referencias a otros trabajos relevantes relacionados con el área de investigación, más concretamente, trabajos de localización y navegación autónoma. Estos trabajos pueden dar una idea de las necesidades básicas que se han de cumplir a la hora de realizar un correcto sistema de localización. Este trabajo queda reflejado y explicado en la Parte II de este documento.

### **Desarrollo del procedimiento de odometría visual monocular**

Dentro del procedimiento planteado se han presentado diversas novedades teniendo siempre como objetivo el aumento de la tasa de muestreo de imágenes, la reducción de los errores en la odometría visual y el mantenimiento del uso de componentes domésticos y de bajo coste.

Las principales contribuciones se encuadran en:

- Extracción de características y elaboración de la estructura física del sistema de visión
- Cálculo del Centro Instantáneo de Rotación
- Procesos de optimización a posteriori

En el proceso de extracción y seguimiento de características se ha encontrado una solución óptima que favorece el correcto funcionamiento y mitiga los errores ocasionados por un fallo en el seguimiento de las diferentes características. Todo ello parte de una estructura física del sistema de visión especial, a la que se llega tras un estudio prolongado en el tiempo y en las pruebas. El sistema de visión consiste en una única cámara enfocada hacia el suelo. De esta forma se logra un campo de visión limitado y finito que permite la fácil obtención de medidas de distancia. Por contra, nos provoca una mayor dificultad a la hora de la extracción y seguimiento de características. Esto es debido a que en el suelo, los puntos característicos no abundan y las texturas son más uniformes que en una imagen enfocando el horizonte.

Se ha utilizado el *tracker* Lukas-Kanade-Tomasi cuyos tiempos de ejecución son claramente inferiores -en estas condiciones- que el de otras

técnicas más novedosas. Esto nos permite mantener la tasa de captura de imágenes alta aunque -debido a la dificultad en la extracción de características, ante una menor diversidad en las texturas dentro del campo de visión- provocará la incorporación de sistemas de filtrado y optimización de resultados.

Se aporta en esta investigación la idea del cálculo del Centro Instantáneo de Rotación basándose exclusivamente en los datos recibidos por el sistema de visión. Esta técnica es novedosa en este entorno. Existe obviamente este concepto en problemas de cinemática partiendo de las características del robot y a partir de los datos obtenidos por sistemas propioceptivos como lo puede ser la odometría de las ruedas. En la literatura hay múltiples ejemplos de esta idea, pero se incorpora aquí la idea del uso del Centro Instantáneo de Rotación basándose exclusivamente en técnicas de flujo óptico. Esta aproximación logra una reducción importante en los cálculos del movimiento respecto a otros procedimientos de cálculo de odometría visual -que basan el cálculo del movimiento en problemas de obtención de la estructura en 3D a partir del movimiento (SFM, *Structure From Motion*).

Por último se realizan procesos de optimización que refinan el cálculo del movimiento realizado. El primer proceso de optimización consiste en la aplicación de simples filtros estadísticos  $(1 - \sigma)$  tanto en el seguimiento de características (en cuanto a módulo y ángulo), como en el cálculo del Centro Instantáneo de Rotación y como en el cálculo del giro realizado. Este tipo de filtrados son computacionalmente poco costosos. Al aplicarse en tres momentos diferentes del proceso, el refinamiento resulta veloz y robusto.

El último proceso de optimización es la creación de un clasificador borroso que diferencia entre cuatro tipos de movimientos: parada, giro, avance en línea recta, ruido. La aplicación de este clasificador es computacionalmente poco costoso y logra una mejoría notable a la hora del cálculo del movimiento final. Principalmente esto se debe a la inexactitud que provoca un movimiento en línea recta -que no deja de ser un movimiento circular con radio infinito- y el estado de parada -que es un movimiento cuyo avance es cero-. Estas inexactitudes que irían acumulando errores se resuelven aplicando el clasificador borroso. De nuevo este concepto es novedoso en problemas de odometría visual.

### **Fusión sensorial odométrica**



Como aplicación del trabajo de investigación, se ha desarrollado una fusión sensorial utilizando los dos sensores odométricos de los que dispone el robot de pruebas: odometría de las ruedas y odometría visual. Esta aplicación busca demostrar que con la fusión de dos sensores odométricos se puede lograr un sistema odométrico más robusto. La localización basada en la odometría de las ruedas es sensible a entornos resbaladizos y rugosos, bloqueos de las ruedas, derrapes, etc. La localización basada en la odometría visual por su parte es sensible a movimientos en la escena de la pantalla, oclusiones de objetos, objetos con relieve en el suelo, etc. Ambos sensores pueden compatibilizarse de forma que por ejemplo, en el caso de un bloqueo de las ruedas, la odometría visual devolvería resultados correctos (estado de parada). Por otra parte, en el caso de movimientos en el campo de visión con el robot en estado de parada, la odometría de las ruedas proporcionarían resultados correctos al no estar en movimiento las ruedas.

Para la fusión sensorial se ha utilizado el Filtro de Kalman Extendido (EKF) cuyo uso para estos objetivos está plenamente demostrado en trabajos anteriores.

### **Resultados y aplicaciones**

Llegados a la hora de analizar los resultados se puede afirmar que se ha desarrollado un sistema de odometría visual monocular con resultados válidos para la navegación autónoma de un robot. Respecto a trabajos previos se ha logrado desarrollar un sistema utilizando exclusivamente componentes domésticos y de bajo coste, como son una cámara y un procesador comerciales. También se ha logrado un procedimiento con un coste computacional suficientemente bajo que permita una tasa de adquisición de imágenes para trabajar bajo restricciones de tiempo real.

Por otra parte, el sistema es completamente independiente de cualquier otro sensor equipado en el robot, de forma que se puede considerar un sensor exteroceptivo puro, al tener como fuente información puramente ajena al robot.

A esto hay que añadir que la principal aplicación o caso de estudio desarrollado en este trabajo ha dado resultados satisfactorios incluso en situaciones extremas en las que uno de los dos sensores odométricos (odometría de ruedas y odometría visual) falla estrepitosamente: en el caso de la odometría de las ruedas, un bloqueo de ruedas; en el caso de la odometría visual, movimientos en la escena continuados.

## 5.2. Trabajo futuro

Existe una línea de trabajo futura abierta claramente. A día de hoy, el sistema se encuentra implantado en un robot de laboratorio que realiza recorridos en un edificio con salidas al exterior en zonas no abruptas en exceso. La velocidad de funcionamiento se encuentra alrededor de los 5 km/h, lo que equivaldría al ritmo de un ser humano andando. La línea de trabajo que queda abierta actualmente y como continuación de la investigación actual es la implantación del sistema desarrollado en un vehículo a motor que transite por una carretera de forma que se pueda observar la escalabilidad del sistema ante situaciones más complejas.

Con ello lo que se buscaría sería encontrar nuevas aplicaciones a la odometría visual, más allá del objetivo de localización de un vehículo. El sistema podría ser utilizado para evaluar diferencias entre la trayectoria calculada a partir de los datos proporcionados por los sensores odométricos de las ruedas -o por cualquier otro método- y la trayectoria calculada por la odometría visual. Con esto se podría decidir por ejemplo si se está produciendo un efecto de subviraje o sobreviraje en una curva.

Para este tipo de problemas es si cabe más importante la reducción del coste computacional del procedimiento, puesto que la frecuencia del muestreo debería aumentar en proporción a la velocidad a la que se mueve el sistema de visión.

De la misma manera, el posicionamiento físico y la orientación del sistema de visión ha de ser estudiado en profundidad. En el trabajo actual, resultó finalmente de suma importancia la posición del sistema de visión y su orientación respecto del suelo, fijando así un campo de visión finito y limitado.



## Apéndice A

# Base teórica de la fusión sensorial

En este apéndice se detalla la base teórica utilizada durante la fusión sensorial de los dos sensores odométricos (odometría visual y odometría de las ruedas). Se utiliza el Filtro de Kalman Extendido para calcular el estado del robot utilizando información proporcionada por ambos sensores.

### A.1. El Filtro de Kalman

El filtro de Kalman es una solución recursiva para el cálculo del estado de un sistema dinámico lineal estocástico. Se basa en la toma de observaciones periódicas en el tiempo que contienen ruido para lograr una estimación del estado del sistema. El estado de un sistema está formado por un conjunto de variables que representan las propiedades del mismo. En el caso de un robot, el estado de un sistema suele contener su posición y orientación, además de otras posibles variables que identifiquen otras propiedades significativas para la funcionalidad del robot, por ejemplo, la velocidad [19].

El algoritmo está formado por dos fases diferenciadas: una fase de predicción y una fase de corrección basada en la observación. Por cada iteración del algoritmo, éste estima los valores de las variables que forman el estado del sistema durante la fase de predicción. Para ello se

sirve de un modelo estadístico explícito de cómo el estado evoluciona en el tiempo (modelo del sistema), que depende del estado anterior. Los valores se tratan con una cierta incertidumbre en la predicción, asumiendo de esta manera que existe un ruido que afecta al modelo del sistema.

Una vez que se tiene la predicción del estado del sistema, se inicia la fase de observación. A partir de una observación, también perturbada por un cierto ruido, y un modelo estadístico explícito de cómo las observaciones están relacionadas con el estado (modelo de observación), la estimación de los valores de las variables que forman parte del estado del sistema son actualizadas en función de unos determinados pesos. El peso irá en función de la certeza de la predicción del valor de cada variable realizada en la primera fase.

Al ser un algoritmo recursivo no es necesario almacenar todas las observaciones que han tenido lugar. Tan sólo será necesario utilizar la observación que tiene lugar en cada iteración y el estado calculado en el instante anterior. Cualquier otra información sobre el estado del sistema en instantes de tiempo anteriores no es necesaria.

El filtro de Kalman es utilizado en multitud de aplicaciones y, concretamente, es utilizado comúnmente en problemas de fusión sensorial.

### **A.1.1. Suposiciones**

El filtro de Kalman basa sus cálculos en un sistema estocástico y lineal, por tanto su comportamiento es intrínsecamente no determinístico. Se asume también que el ruido que afecta al sistema cumple ciertas características. Para el correcto funcionamiento del filtro se ha de disponer de un modelo del sistema así como de un modelo de la observación.

#### **A.1.1.1. Sistema estocástico lineal**

Un sistema estocástico es aquel cuyo comportamiento es no determinístico. El estado actual del sistema viene determinado por el estado anterior, las acciones predecibles que han tenido lugar y elementos aleatorios no determinísticos que perturben dichas acciones. En definitiva, cualquier sistema analizable probabilísticamente se puede considerar un sistema estocástico.

En los problemas de robótica, se debe asumir además que los sensores introducen ruido en las observaciones, puesto que no existe ningún sensor perfecto. Así por tanto, el filtro de Kalman se basa en modelos lineales que incorporan las perturbaciones por ruido tanto del propio sistema como de las observaciones de los sensores.

Sin embargo, muchos sistemas del mundo real contienen no linealidades. Para ello, es necesario linealizar dichos modelos mediante el Filtro Extendido de Kalman.

#### A.1.1.2. Modelo del Sistema

El modelo del sistema describe la forma en la que evoluciona el estado real del sistema en el tiempo. Este modelo es utilizado por el filtro de Kalman para realizar la fase de predicción del estado del sistema en el próximo instante de tiempo. Si se representa en el espacio de estados para un sistema discreto, la evolución del estado  $x$  se describe como:

$$\begin{aligned} x_{k+1} &= F_k x_k + B_k u_k + v_k \\ v_k &\sim \mathcal{N}(0, Q_k) \end{aligned} \tag{A.1}$$

donde  $x_k$  es la variable de estado  $n$ -dimensional en el instante de tiempo  $k$ ,  $F_k$  es la matriz de transición de estados,  $u_k$  es la entrada de control al sistema y  $v_k$  representa el ruido del sistema.

El ruido del sistema se considera un ruido blanco gaussiano. De esta manera el ruido se puede modelar mediante una distribución normal multivariante con media  $E[v(k)] = 0$  y covarianza  $Q_k$ , cumpliéndose que  $E[v_k(i) v_k^T(j)] = \delta_{ij} Q_k(i)$ .

#### A.1.1.3. Modelo de Observación

El modelo de observación describe la relación existente entre las observaciones y el estado del sistema. Se utiliza en la fase de corrección, de forma que cada vez que se produce una observación se corrige la predicción del estado realizada a partir del modelo del sistema expuesto anteriormente.

El modelo de observación permite predecir el valor de una observación partiendo de un estado concreto y así compararla con el verdadero valor

de la observación tomado en ese instante. A partir de esa comparación se podrá calcular el error de la observación, que será la base para la corrección de la estimación del estado.

El filtro de Kalman supone que las observaciones se pueden modelar mediante una ecuación que relacione linealmente el estado del sistema con las observaciones, de modo que para un sistema discreto, el modelo de observación se representa de la siguiente manera:

$$\begin{aligned} z_k &= H_k x_k + w_k \\ w_k &\sim \mathcal{N}(0, R_k) \end{aligned} \quad (\text{A.2})$$

donde  $z_k$  es un vector de observaciones  $m$ -dimensional,  $H_k$  es la matriz de observación,  $x_k$  es el vector de estado obtenido en la fase de predicción a partir del modelo del sistema, y  $w_k$  es el ruido en la observación. El valor de error en la medida incorporado en el modelo de observación es una perturbación por ruido blanco gaussiano con media  $E[w(k)] = 0$  y covarianza  $R_k$ , cumpliéndose que  $E[w_k(i) w_k^T(j)] = \delta_{ij} R_k(i)$ .

#### A.1.1.4. Características del ruido

El filtro de Kalman supone que tanto el ruido en el modelo del sistema,  $v(k)$ , como el ruido en el modelo de observación,  $w(k)$ , son variables aleatorias multivariantes e independientes entre sí que modelan ruido blanco gaussiano:

- Al asumir que las fuentes de ruido son independientes significa que el ruido de una determinada variable en un instante de tiempo no afecta al resto de variables del sistema. Esto significa que el ruido en la observación tomada a partir de los encoders de las ruedas no se ve influenciado por el ruido a la hora de la captura de las imágenes para el cálculo de la odometría visual.
- El ruido blanco es una perturbación aleatoria de manera que dos valores de ruido en dos instantes de tiempo diferentes no guardan ninguna correlación estadística. De esta manera, no se puede predecir el valor de ruido en un instante de tiempo a partir de los valores de ruido anteriores. Una señal de ruido blanco contiene todas las frecuencias y todas ellas guardan la misma potencia, esto

es, la densidad espectral de potencia es constante. El ruido tiene media cero, lo que significa que el ruido no es sistémico sino que los valores de ruido son aleatorios. Un ejemplo de ruido sistémico sería un error a la hora de medir el radio de una rueda. En los cálculos de odometría de las ruedas a partir de los encoders, el resultado estaría sesgado constantemente por ese dato mal medido, por lo que la media del ruido sería distinta de cero.

- Una de las suposiciones tenidas en cuenta es que la distribución del ruido se gaussiana, apoyándose en el teorema central de límite en base a que los ruidos modelados son el resultado de la suma de un gran número de fuentes de error aleatorias. Por otra parte, al calibrar los sensores es normalmente sencillo realizar experimentos para averiguar su media y varianza del error. La función de densidad de probabilidad de una distribución gaussiana está completamente descrita por su media y su varianza, por lo que no se necesitan calcular estadísticos de orden mayor y se captura toda la información disponible acerca del ruido.

Siendo el ruido  $v(k)$  un ruido blanco gaussiano, al ser éste incorporado al modelo del sistema (A.1), se logra que el estado  $s_k$  pase a responder como una distribución normal. Esto se debe a que una de las propiedades de las distribuciones normales es que dadas dos variables  $x_1 \sim \mathcal{N}(\mu_1, \Sigma_1)$  y  $x_2 \sim \mathcal{N}(\mu_2, \Sigma_2)$  aleatorias e independientes con una distribución normal multivariante, la suma de ambas seguirá teniendo una distribución normal multivariante:

$$x_1 + x_2 \sim \mathcal{N}(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2) \quad (\text{A.3})$$

Por tanto, asumiendo que  $v_k \sim \mathcal{N}(0, Q_k)$  es parte de la ecuación y representa el ruido, se puede considerar que el resto de la expresión carece de ruido, esto es, la covarianza es 0. Por tanto, se puede considerar que  $(F_k x_k + B_k u_k) \sim \mathcal{N}(F_k x_k + B_k u_k, 0)$  y utilizando la propiedad (A.3), resulta que:

$$x_{k+1} \sim \mathcal{N}(F_k x_k + B_k u_k, Q_k) \quad (\text{A.4})$$

De la misma manera se puede llegar a la conclusión de que el modelo de observación (A.2) responde a la distribución normal:



$$z_k \sim \mathcal{N}(H_k x_k, R_k) \quad (\text{A.5})$$

### A.1.2. Formulación del problema de estimación

Una vez que se tiene modelado el sistema mediante la ecuación (A.1) y descritas las observaciones y su relación con el sistema mediante la ecuación (A.2), el filtro de Kalman buscará efectuar una estimación ( $\hat{x}$ ) del valor real del estado ( $x$ ). El error de esa estimación del estado ( $\tilde{x}$ ) será la diferencia entre el valor real y el estimado:

$$\tilde{x} = x - \hat{x} \quad (\text{A.6})$$

El filtro de Kalman no sólo calcula una estimación del estado ( $\hat{x}$ ) a partir del modelo del sistema, sino que también mantiene una medida estadística de la incertidumbre asociada a esa estimación. Esta medida de incertidumbre se calcula como la matriz de covarianza de  $\tilde{x}$ , y se representa por  $P$ . Por lo tanto, el filtro de Kalman estima el valor real del estado como  $x \sim \mathcal{N}(\hat{x}, P)$ , es decir, como una función de densidad de probabilidad normal multivariante con media  $\hat{x}$  y covarianza  $P$ , donde:

$$\hat{x} = E[x] = \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_n \end{bmatrix} \quad (\text{A.7})$$

$$P = E[\tilde{x} \cdot \tilde{x}^T] = E[(x - \hat{x})(x - \hat{x})^T] = \begin{pmatrix} P_{11} & \dots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \dots & P_{nn} \end{pmatrix} \quad (\text{A.8})$$

En el contexto de la localización de un robot móvil, el hecho de que el filtro de Kalman estime el estado como una distribución de probabilidad gaussiana implica que la ubicación del robot no viene dada como una única posición, sino que se expresa como una distribución de posiciones posibles del robot, cada una de ellas asociada a una probabilidad concreta.

### A.1.3. Algoritmo

Como se ha descrito en los apartados anteriores, el filtro de Kalman calcula la estimación del estado ( $\hat{x}$ ) basándose en la iteración de ciclos con dos fases: predicción y corrección. La primera fase calcula una predicción del estado utilizando el modelo del sistema (A.1). La segunda fase utiliza el modelo de la observación (A.2) para, a partir de una observación disponible en un instante de tiempo, corregir la predicción del estado.

La notación  $\hat{x}_{k_1|k_2}$  representa la estimación del estado en el instante de tiempo  $k_1$  teniendo en cuenta las observaciones disponibles en el instante de tiempo  $k_2$  incluido. A continuación se muestran las ecuaciones para cada una de las fases sin su demostración matemática correspondiente. La obtención de las ecuaciones viene claramente detallada y especificada en la literatura.

**PREDICCIÓN:** En cada instante de tiempo, el estado del sistema puede variar y por ello en cada incremento de tiempo, el filtro de Kalman calcular una nueva estimación del estado a partir de la última estimación disponible. Dada la estimación del sistema y su matriz de covarianza ( $\hat{x}_{k|k}, P_{k|k}$ ) en el instante de tiempo  $k$ , la predicción ( $\hat{x}_{k+1|k}, P_{k+1|k}$ ) en el instante de tiempo  $k+1$  se calculan como:

$$\hat{x}_{k+1|k} = F_k \cdot \hat{x}_{k|k} + B_k \cdot u_k \quad (\text{A.9})$$

$$P_{k+1|k} = F_k \cdot P_{k|k} \cdot F_k^T + Q_k \quad (\text{A.10})$$

**CORRECCIÓN:** A diferencia de la fase de predicción, la fase de corrección solamente se efectúa cuando están disponibles nuevas observaciones, que proporcionan información directa acerca del estado actual del sistema. Las ecuaciones de corrección actualizan la estimación del estado más reciente mediante la incorporación de la información ganada a partir de las observaciones.

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \cdot \tilde{z}_{k+1} \quad (\text{A.11})$$

$$P_{k+1|k+1} = [I - K_{k+1} \cdot H_{k+1}] \cdot P_{k+1|k} \quad (\text{A.12})$$

donde (A.11) es la corrección de la estimación del estado, (A.12) es la corrección de la covarianza del estado, y

$$\tilde{z}_{k+1} = z_{k+1} - H_{k+1} \cdot \hat{x}_{k+1|k} \quad (\text{A.13})$$

$$S_{k+1} = H_{k+1} \cdot P_{k+1|k} \cdot H_{k+1}^T + R_{k+1} \quad (\text{A.14})$$

$$K_{k+1} = P_{k+1|k} \cdot H_{k+1}^T \cdot S_{k+1}^{-1} \quad (\text{A.15})$$

(A.13) es la innovación, (A.14) es la covarianza de la innovación y (A.15) es la ganancia óptima de Kalman, que está elegida de modo que garantiza que la estimación del estado minimiza el error cuadrático medio.

Un aspecto a destacar en la ecuación de innovación (A.13) es que el filtro de Kalman efectúa una predicción de la observación definida mediante la expresión  $H_{k+1} \cdot \hat{x}_{k+1|k} = \hat{z}_{k+1}$ . Esta predicción indica cuál es el valor que el filtro predice que debería tener la observación  $z_{k+1}$ , basándose en la predicción del estado  $\hat{x}_{k+1|k}$ . Reescribiendo la ecuación de innovación como  $\tilde{z}_{k+1} = z_{k+1} - \hat{z}_{k+1}$  resulta más evidente que la innovación es una medida del error que el filtro comete al predecir la observación.

Cuando la innovación es próxima a cero, entonces la predicción de la observación es muy similar a la observación real, lo que implicaría que la predicción del estado (desde la que se hizo la predicción de la observación) sería ya bastante precisa y no necesitaría una gran corrección. Por el contrario, cuando la innovación es grande, la predicción de la observación difiere bastante de la observación real. Esto quiere decir que la predicción del estado necesitaría una fuerte corrección.

## A.2. El Filtro de Kalman Extendido (EKF)

Cuando los sistemas con los que se trabaja son robots móviles, suponer que el modelo de sistema y el modelo de observación son lineales con perturbaciones por ruido blanco gaussiano son unos requisitos que difícilmente se pueden cumplir. Basta pensar en un robot con velocidad

constante de avance y de rotación para darse cuenta que la trayectoria que describe es circular y no se puede describir con un modelo de sistema lineal. Esto hace que el filtro de Kalman lineal resulte inaplicable a la mayoría de problemas de localización de robots, a excepción de los más triviales.

A la hora de aplicar el filtro de Kalman a problemas no lineales existen principalmente dos técnicas. La primera consiste en utilizar una serie de Taylor de primer orden en torno a una trayectoria de referencia estática, dando como resultado lo que se conoce como *Filtro de Kalman Linealizado*. Es una técnica útil cuando existe una gran cantidad de información a priori acerca del problema. El principal problema de este enfoque es que se utiliza la misma trayectoria de referencia mientras dura el proceso de estimación, y los errores crecerían sin límite en el caso de que haya una gran diferencia entre el estado del sistema y la trayectoria de referencia. Para evitar este problema, la solución consiste en linealizar mediante una serie de Taylor en torno a una trayectoria de referencia *dinámica* que se está actualizando constantemente con estimaciones del estado que son producto de las observaciones. Es decir, cada vez que se calcula una nueva estimación del estado, se recalcula una nueva trayectoria de referencia mejor y más precisa y se incorpora al proceso de estimación. Cuando se utiliza una serie de Taylor de primer orden, el filtro resultante se conoce como *Filtro Extendido de Kalman* (EKF), mientras que si se utiliza una serie de Taylor de segundo orden se obtiene el *EKF de segundo orden*. En este trabajo se utilizará exclusivamente el EKF estándar o de primer orden, basados en la suposición de un modelo de sistema no lineal y un modelo de observación no lineal, cuyas ecuaciones se describen a continuación.

### A.2.1. Modelo del Sistema

En este trabajo, el EKF se utiliza para fusionar la información obtenida a partir de los encoders de las ruedas con la información odométrica obtenida a partir del sensor de visión que se encuentra en el robot.

Para una correcta definición tanto del problema a tratar como de la solución aportada, se pasa ahora a describir el modelo del sistema, el vector de estado, la función de transición de estados y el ruido del sistema.

#### A.2.1.1. Modelo del sistema

En el EKF, el sistema tiene que estar representado por una ecuación *no lineal* de la forma:

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k \\ \mathbf{v}_k &\sim \mathcal{N}(0, \mathbf{Q}_k)\end{aligned}\tag{A.16}$$

donde  $\mathbf{f}(\cdot)$  es una *función de transición de estados no lineal* y  $\mathbf{v}_k$  representa una perturbación por ruido blanco gaussiano, con media  $E[\mathbf{v}(k)] = 0$  y covarianza  $\mathbf{Q}_k$ , cumpliéndose que  $E[\mathbf{v}_k(i)\mathbf{v}_k^T(j)] = \delta_{ij}\mathbf{Q}_k(i)$ .

#### A.2.1.2. Vector de estado

El vector de estado se ha definido empleando variables que permitan localizar al robot en el espacio de manera única. Considerando un sistema de referencia global cartesiano bidimensional, el estado se ha determinado mediante la posición del robot de la forma  $(x, y, \theta)$ , y la velocidad del robot de la forma  $(\dot{x}, \dot{y}, \dot{\theta})$ .

Así, utilizando un sistema de tiempos discreto, el estado del robot ( $\mathbf{x}$ ) en el instante de tiempo  $k$  está dado por:

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ \dot{x}_k \\ \dot{y}_k \\ \dot{\theta}_k \end{bmatrix}\tag{A.17}$$

En el modelo del sistema, los cambios en el estado del robot, es decir, cambios de posición y velocidad, se miden mediante encoders incrementales, que proporcionan medidas de desplazamiento relativas con respecto al sistema de referencia local del robot. Los incrementos de posición interpretados de manera aislada no resultan de ninguna utilidad. Sin embargo, si se integran en el tiempo, permiten conocer la trayectoria del robot en el sistema de referencia global, siempre y cuando se conozca la posición de partida. Para llevar a cabo esta tarea, y convertir los

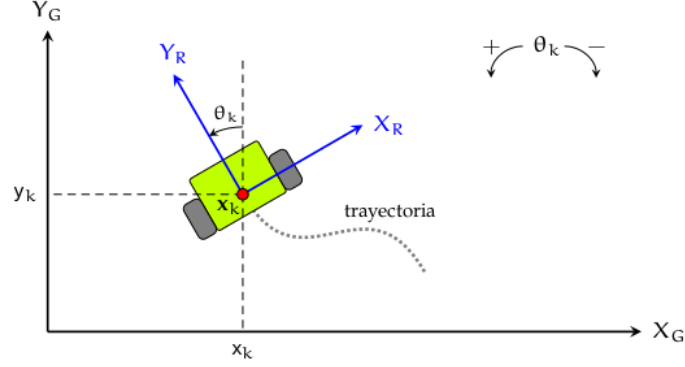


Figura A.1: Relación entre el sistema de referencia local del robot y el sistema de referencia global

giros o traslaciones medidos en el sistema de referencia local del robot  $\{\mathbf{X}_R, \mathbf{Y}_R\}$  al sistema de referencia global  $\{\mathbf{X}_G, \mathbf{Y}_G\}$

Como punto identificativo  $(x_k, y_k)$  del robot, coincidente con el origen del sistema de referencia local, se ha elegido el punto medio del eje de las ruedas motrices, apuntando el eje  $\mathbf{Y}_R$  en la dirección de avance del robot. La orientación del robot  $(\theta_k)$  se ha definido como el ángulo formado por el eje  $\mathbf{Y}_R$  con respecto a  $\mathbf{Y}_G$ , tomando como positivo el giro en el sentido contrario a las agujas del reloj. El origen del sistema de referencia global se debe elegir arbitrariamente en algún punto del espacio cartesiano.

#### A.2.1.3. Función de transición de estados

En el EKF, la función de transición de estados  $\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$  es una función no lineal que, conociendo la entrada de control  $(\mathbf{u}_k)$  que ha modificado el estado del sistema, permite calcular el nuevo estado del sistema  $(\mathbf{x}_{k+1})$  a partir del estado actual  $(\mathbf{x}_k)$ . En el caso de nuestro problema de localización, esta función coincide con el modelo odométrico de nuestro robot diferencial, y se utiliza para calcular la nueva posición del robot a partir del desplazamiento relativo medido por sus encoders.

El vector  $\mathbf{u}_k$  es el resultado de la acción de control medido mediante los encoders del robot, y representa el desplazamiento del robot en su sistema de referencia local al pasar del instante de tiempo  $k$  al instante  $k + 1$ :

$$\mathbf{u}_k = \begin{bmatrix} \Delta S_{k,i} \\ \Delta S_{k,d} \end{bmatrix} = \begin{bmatrix} \frac{\Delta N_i}{n} \cdot C_i \\ \frac{\Delta N_d}{n} \cdot C_d \end{bmatrix} \quad (\text{A.18})$$

donde  $\Delta N_i$  y  $\Delta N_d$  indican el número de ticks contados por el encoder izquierdo y derecho respectivamente durante el periodo de muestreo,  $n$  es la resolución del encoder en ticks por vuelta y  $C$  es el perímetro de las ruedas.

Al transformar los desplazamientos relativos del vector  $\mathbf{u}_k$  del sistema de referencia local al global se obtiene la función de transición de estados:

$$\hat{\mathbf{x}}_{k+1|k} = f(\hat{\mathbf{x}}_k, \mathbf{u}_k) = \begin{bmatrix} x_k + \Delta S_k \cdot \sin\left(\theta_k + \frac{\Delta\theta_k}{2}\right) \\ y_k + \Delta S_k \cdot \cos\left(\theta_k + \frac{\Delta\theta_k}{2}\right) \\ \theta_k + \Delta\theta_k \\ \Delta S_k \cdot \sin\left(\frac{\Delta\theta_k}{2}\right) \\ \Delta S_k \cdot \cos\left(\frac{\Delta\theta_k}{2}\right) \\ \Delta\theta_k \end{bmatrix} \quad (\text{A.19})$$

donde  $b$  es la distancia entre las ruedas motrices del robot, y los términos  $\Delta S_k$  y  $\Delta\theta_k$  representan el desplazamiento y el cambio de orientación del robot en su sistema de referencia local. Ambos se pueden calcular mediante las expresiones:

$$\Delta S_k = \frac{\Delta S_{k,i} + \Delta S_{k,d}}{2} \quad (\text{A.20})$$

$$\Delta\theta_k = \frac{\Delta S_{k,i} - \Delta S_{k,d}}{b} \quad (\text{A.21})$$

#### A.2.1.4. Ruido del sistema

Existen dos causas principales que introducen errores en los cálculos del modelo odométrico representado en la función de transición de estados:

- Cuando la velocidad angular del robot es demasiado alta en comparación con el periodo de muestreo, el modelo odométrico no es perfecto y se producen pequeños errores en los cálculos. Esto se debe a que el modelo utiliza la aproximación para ángulos pequeños  $\sin(\theta) \approx \theta$ .
- Los encoders, al igual que cualquier otro sensor, proporcionan medidas que están sujetas a errores. Así, al calcular la distancia que se ha desplazado cada rueda (A.18), el resultado obtenido es una aproximación del valor real. En este caso, los errores se pueden ver aumentados por problemas propios del entorno donde el robot esté realizando las tareas, de forma que un suelo excesivamente rugoso o resbaladizo puede provocar errores en la medida de la odometría de las ruedas.

Estas dos fuentes de ruido se interpretan en conjunto como la incertidumbre que afecta a la distancia que se ha desplazado cada rueda durante el periodo de muestreo. Dicha incertidumbre se ha decidido modelar como un valor proporcional al valor absoluto de la distancia recorrida por cada rueda, de modo que la matriz de covarianza del sistema se expresa como:

$$\mathbf{Q}_k = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_d^2 \end{bmatrix} \quad (\text{A.22})$$

donde

$$\begin{aligned} \sigma_i &= k_i \cdot |\Delta S_{k,i}| \\ \sigma_d &= k_d \cdot |\Delta S_{k,d}| \end{aligned} \quad (\text{A.23})$$

y los términos  $\Delta S_{k,i}$  y  $\Delta S_{k,d}$  son las distancias recorridas por las ruedas izquierda y derecha, respectivamente, mientras que  $k_i$  y  $k_d$  son constantes que recogen la incertidumbre producida por las dos fuentes de error citadas. Los valores para las constantes de error  $k_i$  y  $k_d$  dependen del robot y del entorno y deben seleccionarse experimentalmente estudiando diferentes movimientos del robot. Se ha supuesto que el ruido que afecta al sistema es ruido blanco gaussiano, de modo que los errores en cada una de las ruedas son independientes, y por ello los elementos que no están en la diagonal de la matriz de covarianza son cero.



### A.2.2. Modelo de Observación

El modelo del sistema se utiliza para la predicción del nuevo estado a partir del estado anterior y del movimiento ocurrido, identificado por el cálculo de la odometría de las ruedas. En la fase de corrección se utiliza el modelo de observación y una medida tomada en un instante de tiempo para corregir la predicción.

En este caso, la medida tomada para corregir el estado estimado será la medida de odometría visual. El resultado de la odometría visual son la velocidad angular  $\omega$  y la velocidad lineal  $v$  que ha tenido lugar entre el instante de tiempo de la medida actual  $(k+1)$  y el instante de tiempo de la medida anterior  $(k)$ .

#### A.2.2.1. Modelo de observación

Igual que en el modelo de sistema, el modelo que relaciona las observaciones con el estado también pasa a estar expresado mediante una ecuación *no lineal*:

$$\begin{aligned} \mathbf{z}_{k+1} &= \mathbf{h}(\hat{\mathbf{x}}_{k+1|k}) + \mathbf{w}_{k+1} \\ \mathbf{w}_{k+1} &\sim \mathcal{N}(0, \mathbf{R}_{k+1}) \end{aligned} \quad (\text{A.24})$$

donde  $\mathbf{h}(\cdot)$  es una *función de observación no lineal* y  $\mathbf{w}_{k+1}$  es una perturbación por ruido blanco gaussiano, con media  $E[\mathbf{w}(k+1)] = 0$  y covarianza  $\mathbf{R}_{k+1}$ , cumpliéndose que  $E[\mathbf{w}_{k+1}(i)\mathbf{w}_{k+1}^T(j)] = \delta_{ij}\mathbf{R}_{k+1}(i)$ .

En nuestro caso, la medida de observación que nos proporciona la odometría visual puede representarse mediante:

$$\mathbf{z}_{k+1} = \begin{bmatrix} \omega \\ \rho \end{bmatrix} \quad (\text{A.25})$$

donde  $\omega$  es la velocidad angular y  $\rho$  es el módulo del vector de velocidad lineal, calculadas a partir de la información de la odometría visual.

#### A.2.2.2. Función de observación

El objetivo de la función de observación es identificar los valores de la predicción que identifican los valores de la observación. En este caso,

la función de observación sobre la predicción realizada ( $\mathbf{h}(\hat{x}_{k|k+1})$ ) se describe de la siguiente manera:

$$\mathbf{h}(\hat{x}_{k+1|k}) = \begin{bmatrix} \dot{\theta}_{k+1|k} \\ \sqrt{\dot{x}_{k+1|k}^2 + \dot{y}_{k+1|k}^2} \end{bmatrix} \quad (\text{A.26})$$

donde  $\dot{\theta}_{k+1|k}$  es la velocidad angular que se toma a partir del modelo odométrico y  $\dot{x}_{k+1|k}$  y  $\dot{y}_{k+1|k}$  son las componentes  $x$  y en  $y$ , respectivamente, del vector velocidad lineal, que se calculan en la fase de predicción.

De esta forma se puede calcular el error de la predicción respecto a la observación como:

$$\tilde{\mathbf{z}}_{k+1} = \mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1|k}) \quad (\text{A.27})$$

De la misma manera que en el modelado del sistema, las observaciones se ven afectadas por un error que puede ser debido a diferentes factores, como por ejemplo, cambios en la iluminación a la hora de la captura de las imágenes, objetos móviles en el campo de visión, etc.

De esta forma, esta incertidumbre ha quedado modelada como un valor proporcional al valor de las velocidades angulares y lineales obtenidas en el proceso del cálculo de la odometría visual, de modo que la matriz de covarianza de la observación se expresa como:

$$\mathbf{R}_{k+1} = \begin{bmatrix} R_\omega & 0 \\ 0 & R_\rho \end{bmatrix} \quad (\text{A.28})$$

donde

$$\begin{aligned} R_\omega &= k_\omega \cdot \omega \\ R_\rho &= k_\rho \cdot \rho \end{aligned} \quad (\text{A.29})$$

y los términos  $k_\omega$  y  $k_\rho$  son constantes que recogen la incertidumbre producidas por las posibles fuentes de error en las imágenes. Los valores para las constantes de error  $k_\omega$  y  $k_\rho$  dependen del entorno fundamentalmente, puesto que los errores que se pueden producir en el cálculo de la odometría visual se deben fundamentalmente a cambios en el entorno.

De esta manera, en un entorno en el que conocemos que los cambios de luminosidad son extremos, la odometría visual producirá más valores erróneos. Igualmente, en un entorno en el que tenemos certeza de que los objetos móviles son escasos, la odometría visual proporciona datos muy precisos. Se ha supuesto que el ruido que afecta a las medidas de odometría visual es un ruido blanco gaussiano, de modo que los errores en la velocidad lineal y en la velocidad angular son independientes. Por ello, los elementos que no están en la diagonal de la matriz de covarianza son cero.

# Bibliografía

- [1] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure: Center surround extremas for realtime feature detection and matching. In *Computer Vision–ECCV 2008*, pages 102–115. Springer, 2008.
- [2] Fred Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183, 1954.
- [3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.
- [4] HR Beom and HS Cho. Mobile robot localization using a single rotating sonar and two passive cylindrical beacons. *Robotica*, 13(03):243–252, 1995.
- [5] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.
- [6] Johann Borenstein. The clapper: a dual-drive mobile robot with internal correction of dead-reckoning errors. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 3085–3090. IEEE, 1994.
- [7] Jean-Yves Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5, 2001.
- [8] Jason Campbell, Rahul Sukthankar, Illah Nourbakhsh, and Aroon Pahwa. A robust visual odometry and precipice detection system

- using consumer-grade monocular vision. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3421–3427. IEEE, 2005.
- [9] Yang Cheng, Mark Maimone, and Larry Matthies. Visual odometry on the mars exploration rovers. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 1, pages 903–910. IEEE, 2005.
- [10] SeungKeun Cho and Jangmyung Lee. Localization of a high-speed mobile robot using global features. *Robotica*, 29(05):757–765, 2011.
- [11] Ondrej Chum and Jiri Matas. Matching with prosac-progressive sample consensus. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 220–226. IEEE, 2005.
- [12] Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, and Micaela Wiseman. Indoor location sensing using geo-magnetism. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 141–154. ACM, 2011.
- [13] Javier Civera, Oscar G Grasa, Andrew J Davison, and JMM Montiel. 1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5):609–631, 2010.
- [14] Andrew I Comport, Ezio Malis, and Patrick Rives. Accurate quadrifocal tracking for robust 3d visual odometry. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 40–45. IEEE, 2007.
- [15] Peter Corke, Dennis Strelow, and Sanjiv Singh. Omnidirectional visual odometry for a planetary rover. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 4007–4012. IEEE, 2004.
- [16] Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410. IEEE, 2003.

- [17] Olivier D Faugeras and Steve Maybank. Motion from point matches: multiplicity of solutions. *International Journal of Computer Vision*, 4(3):225–246, 1990.
- [18] C Fernández-Caramés, V Moreno, B Curto, JF Rodríguez-Aragón, and FJ Serrano. A real-time door detection system for domestic robotic navigation. *Journal of Intelligent & Robotic Systems*, pages 1–18, 2013.
- [19] Carlos Fernández Caramés. Técnicas de navegación para un robot móvil utilizando sistemas de razonamiento espacial. 2012.
- [20] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [21] Wolfgang Förstner. A feature based correspondence algorithm for image matching. *International Archives of Photogrammetry and Remote Sensing*, 26(3):150–166, 1986.
- [22] Giovanni Forzieri, Luca Tanteri, Gabriele Moser, and Filippo Catani. Mapping natural and urban environments using airborne multi-sensor ads40–mivis–lidar synergies. *International Journal of Applied Earth Observation and Geoinformation*, 23:313–323, 2013.
- [23] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part ii: Matching, robustness, optimization, and applications. *Robotics & Automation Magazine, IEEE*, 19(2):78–90, 2012.
- [24] Friedrich Fraundorfer, Davide Scaramuzza, and Marc Pollefeys. A constricted bundle adjustment parameterization for relative scale estimation in visual odometry. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1899–1904. IEEE, 2010.
- [25] Friedrich Fraundorfer, Petri Tanskanen, and Marc Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In *Computer Vision–ECCV 2010*, pages 269–282. Springer, 2010.
- [26] Miguel Angel Garcia and Agusti Solanas. 3d simultaneous localization and modeling from stereo vision. In *Robotics and Automation*,

2004. *Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 847–853. IEEE, 2004.
- [27] Christopher Geyer and Henrik Stewenius. A nine-point algorithm for estimating para-catadioptric fundamental matrices. In *CVPR*, 2007.
  - [28] Roland Goecke, Akshay Asthana, Niklas Pettersson, and Lars Pettersson. Visual vehicle egomotion estimation using the fourier-mellin transform. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 450–455. IEEE, 2007.
  - [29] Rafael C Gonzalez and Richard E Woods. *Digital image processing*, 2002.
  - [30] Ramón González, Mirko Fiacchini, José Luis Guzmán, Teodoro Álamo, and Francisco Rodríguez. Robust tube-based predictive control for mobile robots in off-road conditions. *Robotics and Autonomous Systems*, 59(10):711–726, 2011.
  - [31] Ramon Gonzalez, Francisco Rodriguez, Jose Luis Guzman, Cedric Pradalier, and Roland Siegwart. Combined visual odometry and visual compass for off-road mobile robots localization. *Robotica*, 30(06):865–878, 2012.
  - [32] José Jesús Guerrero, Ruben Martinez-Cantin, and Carlos Sagüés. Visual map-less navigation based on homographies. *Journal of Robotic Systems*, 22(10):569–581, 2005.
  - [33] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.
  - [34] Christopher G Harris and JM Pike. 3d positional integration from image sequences. *Image and Vision Computing*, 6(2):87–90, 1988.
  - [35] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
  - [36] Zhu Hengzhou, Liu Fuqiang, and Zhou Hao. Indoor location service based on fingerprinting and distance relative attenuation model. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2014 Sixth International Conference on*, pages 341–344. IEEE, 2014.

- [37] Wenyao Ho, Asim Smailagic, Daniel P Siewiorek, and Christos Faloutsos. An adaptive two-phase approach to wifi location sensing. In *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pages 5–pp. IEEE, 2006.
- [38] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, and James Collins. Global positioning system. theory and practice. *Global Positioning System. Theory and practice.*, by Hofmann-Wellenhof, B.; Lichtenegger, H.; Collins, J.. Springer, Wien (Austria), 1993, 347 p., ISBN 3-211-82477-4, Price DM 79.00. ISBN 0-387-82477-4 (USA)., 1, 1993.
- [39] Berthold K Horn and Brian G Schunck. Determining optical flow. In *1981 Technical Symposium East*, pages 319–331. International Society for Optics and Photonics, 1981.
- [40] Odile Horn and M Kreutner. Smart wheelchair perception using odometry, ultrasound sensors, and camera. *Robotica*, 27(02):303–310, 2009.
- [41] Andrew Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3946–3952. IEEE, 2008.
- [42] Thomas S Huang and Arun N Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 82(2):252–268, 1994.
- [43] Eagle S Jones and Stefano Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4):407–430, 2011.
- [44] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.
- [45] Qifa Ke and Takeo Kanade. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–390. IEEE, 2003.



- [46] Laurent Kneip, Margarita Chli, Roland Siegwart, Roland Yves Siegwart, and Roland Yves Siegwart. Robust real-time visual odometry with a single camera and an imu. In *BMVC*, pages 1–11, 2011.
- [47] Kurt Konolige. Small vision systems: Hardware and implementation. In *Robotics Research*, pages 203–212. Springer, 1998.
- [48] Kurt Konolige, Motilal Agrawal, and Joan Sola. Large-scale visual odometry for rough terrain. In *Robotics Research*, pages 201–212. Springer, 2011.
- [49] Josef Krames. Zur ermittlung eines objektes aus zwei perspektiven. *Monatshefte für Mathematik und Physik*, 49(1):327–354, 1941.
- [50] Simon Lacroix, Anthony Mallet, Raja Chatila, and Laurent Gallo. Rover self localization in planetary-like environments. In *Artificial Intelligence, Robotics and Automation in Space*, volume 440, page 433, 1999.
- [51] Maxime Lhuillier. Automatic scene structure and camera motion using a catadioptric system. *Computer Vision and Image Understanding*, 109(2):186–203, 2008.
- [52] Bojian Liang and Nick Pears. Visual navigation using planar homographies. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 1, pages 205–210. IEEE, 2002.
- [53] John Lim, Nick Barnes, and Hongdong Li. Estimating relative camera motion from the antipodal-epipolar constraint. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(10):1907–1914, 2010.
- [54] Peng Lin, Qingbin Li, Qixiang Fan, Xiangyou Gao, and Senying Hu. A real-time location-based services system using wifi fingerprinting algorithm for safety risk assessment of workers in tunnels. *Mathematical Problems in Engineering*, 2014, 2014.
- [55] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, MA Fischler and O. Firschein, eds, pages 61–62, 1987.

- [56] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [57] DavidG. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [58] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.
- [59] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007.
- [60] Ameesh Makadia, Christopher Geyer, and Kostas Daniilidis. Correspondence-free structure from motion. *International Journal of Computer Vision*, 75(3):311–327, 2007.
- [61] Larry Matthies and Steven A Shafer. Error modeling in stereo navigation. *Robotics and Automation, IEEE Journal of*, 3(3):239–248, 1987.
- [62] Larry Henry Matthies. Dynamic stereo vision. 1989.
- [63] Paul McIlroy, Edward Rosten, Simon Taylor, and Tom Drummond. Deterministic sample consensus with multiple match hypotheses. In *BMVC*, pages 1–11. Citeseer, 2010.
- [64] Annalisa Milella and Roland Siegwart. Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In *Computer Vision Systems, 2006 ICVS'06. IEEE International Conference on*, pages 21–21. IEEE, 2006.
- [65] Michael J Milford and Gordon F Wyeth. Single camera vision-only slam on a suburban road network. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3684–3689. IEEE, 2008.
- [66] Michael J Milford, Gordon F Wyeth, and David Prasser. Ratslam: a hippocampal model for simultaneous localization and mapping. In *Robotics and Automation, 2004. Proceedings. ICRA'04*.

- 2004 *IEEE International Conference on*, volume 1, pages 403–408. IEEE, 2004.
- [67] Hans P Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document, 1980.
- [68] Hans P. Morevec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'77*, pages 584–584, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc.
- [69] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Real time localization and 3d reconstruction. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 363–370. IEEE, 2006.
- [70] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3565–3572. IEEE, 2007.
- [71] Oleg Naroditsky, Xun S Zhou, Jean Gallier, Stergios I Roumeliotis, and Kostas Daniilidis. Two efficient solutions for visual odometry using directional correspondence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):818–824, 2012.
- [72] David Nistér. An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):756–770, 2004.
- [73] David Nistér. Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329, 2005.
- [74] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–652. IEEE, 2004.
- [75] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, 2006.

- [76] Clark F Olson, Larry H Matthies, Marcel Schoppers, and Mark W Maimone. Robust stereo ego-motion for long distance navigation. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 453–458. IEEE, 2000.
- [77] Clark F Olson, Larry H Matthies, Marcel Schoppers, and Mark W Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229, 2003.
- [78] Diego Ortín and JMM Montiel. Indoor robot motion based on monocular images. *Robotica*, 19(03):331–342, 2001.
- [79] Ignacio Parra, MA Sotelo, David Fernández Llorca, and Manuel Ocaña. Robust visual odometry for vehicle localization in urban environments. *Robotica*, 28(03):441–452, 2010.
- [80] Johan Philip. A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. *The Photogrammetric Record*, 15(88):589–599, 1996.
- [81] Oscar Pizarro, Ryan Eustice, and Hanumant Singh. Relative pose estimation for instrumented, calibrated imaging platforms. In *DICTA*, pages 601–612. Citeseer, 2003.
- [82] Alberto Pretto, Emanuele Menegatti, and Enrico Pagello. Omnidirectional dense large-scale mapping and navigation based on meaningful triangulation. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3289–3296. IEEE, 2011.
- [83] Rahul Raguram, J-M Frahm, and Marc Pollefeys. Exploiting uncertainty in random sample consensus. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2074–2081. IEEE, 2009.
- [84] Jun Rekimoto, Takashi Miyaki, and Takaaki Ishizawa. Lifetag: Wifi-based continuous location logging for life pattern analysis. In *LoCA*, volume 2007, pages 35–49, 2007.
- [85] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

- [86] Davide Scaramuzza. *Omnidirectional vision: from calibration to robot motion estimation*. PhD thesis, Citeseer, 2008.
- [87] Davide Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International journal of computer vision*, 95(1):74–85, 2011.
- [88] Davide Scaramuzza. Performance evaluation of 1-point-ransac visual odometry. *Journal of Field Robotics*, 28(5):792–811, 2011.
- [89] Davide Scaramuzza, Andrea Censi, and Kostas Daniilidis. Exploiting motion priors in visual odometry for vehicle-mounted cameras with non-holonomic constraints. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4469–4476. IEEE, 2011.
- [90] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *Robotics & Automation Magazine, IEEE*, 18(4):80–92, 2011.
- [91] Davide Scaramuzza, Friedrich Fraundorfer, and Roland Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4293–4299. IEEE, 2009.
- [92] Davide Scaramuzza and Roland Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *Robotics, IEEE Transactions on*, 24(5):1015–1026, 2008.
- [93] Adam Schmidt, Marek Kraft, and Andrzej Kasiński. An evaluation of image feature detectors and descriptors for robot navigation. In *Computer Vision and Graphics*, pages 251–259. Springer, 2010.
- [94] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.
- [95] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.

- [96] Kristy Sim and Richard Hartley. Recovering camera motion using linfty minimization. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1230–1237. IEEE, 2006.
- [97] Henrik Stewenius, Christopher Engels, and David Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006.
- [98] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Real-time monocular slam: Why filter? In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2657–2664. IEEE, 2010.
- [99] Peter Sturm, Srikumar Ramalingam, Jean-Philippe Tardif, Simone Gasparini, and João Barreto. Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(1–2):1–183, 2011.
- [100] Rud Sturm. Das problem der projectivität und seine anwendung auf die flächen zweiten grades. *Mathematische Annalen*, 1(4):533–574, 1869.
- [101] Niko Sünderhauf, Kurt Konolige, Simon Lacroix, and Peter Protzel. Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle. In *Autonome Mobile Systeme 2005*, pages 157–163. Springer, 2006.
- [102] Loreto Susperregi, Jose Maria Martínez-Otzeta, Ander Ansuategui, Aitor Ibarguren, and Basilio Sierra. Rgb-d, laser and thermal sensor fusion for people following in a mobile robot. *Int. J. Adv. Robot. Syst*, 2013.
- [103] Tomáš Svoboda and Tomáš Pajdla. Epipolar geometry for central catadioptric cameras. *International Journal of Computer Vision*, 49(1):23–37, 2002.
- [104] J-P Tardif, Michael George, Michel Laverne, Alonzo Kelly, and Anthony Stentz. A new approach to vision-aided inertial navigation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4161–4168. IEEE, 2010.

- [105] J-P Tardif, Yanis Pavlidis, and Kostas Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2531–2538. IEEE, 2008.
- [106] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.
- [107] Philip HS Torr and David W Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International journal of computer vision*, 24(3):271–300, 1997.
- [108] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.
- [109] Bill Triggs. Routines for relative pose of two calibrated cameras from 5 points. 2000.
- [110] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008.
- [111] Hui Wang, Kui Yuan, Wei Zou, and Qingrui Zhou. Visual odometry based on locally planar ground assumption. In *Information Acquisition, 2005 IEEE International Conference on*, pages 6–pp. IEEE, 2005.
- [112] S Yaseen, A Al-Habaibeh, D Su, and F Otham. Real-time crowd density mapping using a novel sensory fusion model of infrared and visual systems. *Safety science*, 57:313–325, 2013.
- [113] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Computer Vision ECCV 1994*, pages 151–158. Springer, 1994.

## **Agradecimiento Oficial**

Esta Tesis Doctoral se ha realizado bajo el amparo y la financiación del Programa de Ayuda a la Contratación de Personal Investigador de Reciente Titulación (Formación de Personal Investigador). Orden EDU/1933/2008, de 11 de noviembre.